

GDCM

3.2.5

Generated by Doxygen 1.16.1

1 GDCM Documentation	1
2 Todo List	3
3 Deprecated List	5
4 Bug List	7
5 Directory Hierarchy	9
5.1 Directories	9
6 Namespace Index	23
6.1 Namespace List	23
7 Hierarchical Index	25
7.1 Class Hierarchy	25
8 Class Index	35
8.1 Class List	35
9 File Index	49
9.1 File List	49
10 Directory Documentation	57
10.1 Common Directory Reference	57
10.2 DataDictionary Directory Reference	59
10.3 DataStructureAndEncodingDefinition Directory Reference	60
10.4 InformationObjectDefinition Directory Reference	61
10.5 MediaStorageAndFileFormat Directory Reference	62
10.6 MessageExchangeDefinition Directory Reference	65
10.7 Python Directory Reference	67
10.8 Source Directory Reference	67
10.9 Utilities Directory Reference	68
10.10 VTK Directory Reference	68
10.11 Wrapping Directory Reference	69
11 Namespace Documentation	71
11.1 gdcml Namespace Reference	71
11.1.1 Detailed Description	86
11.1.2 Typedef Documentation	86
11.1.2.1 AECComp	86
11.1.2.2 ASComp	86
11.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER	86
11.1.2.4 CSCComp	86

11.1.2.5 DAComp	87
11.1.2.6 DTComp	87
11.1.2.7 FileList	87
11.1.2.8 IconImage	87
11.1.2.9 LOComp	87
11.1.2.10 LTCComp	87
11.1.2.11 MacroEntry	87
11.1.2.12 NestedMacroEntries	87
11.1.2.13 PNComp	88
11.1.2.14 SHComp	88
11.1.2.15 STComp	88
11.1.2.16 TMComp	88
11.1.2.17 UCComp	88
11.1.2.18 UIComp	88
11.1.2.19 URComp	88
11.1.2.20 UTComp	88
11.1.3 Enumeration Type Documentation	89
11.1.3.1 CompOperators	89
11.1.3.2 ECharSet	89
11.1.3.3 ENQueryType	90
11.1.3.4 EQueryLevel	90
11.1.3.5 EQueryType	90
11.1.3.6 ERootType	90
11.1.3.7 LodModeType	91
11.1.4 Function Documentation	91
11.1.4.1 add1()	91
11.1.4.2 backslash()	91
11.1.4.3 Clamp()	91
11.1.4.4 clean()	92
11.1.4.5 doround()	92
11.1.4.6 GetVRFromTag()	92
11.1.4.7 operator”!=() [1/2]	92
11.1.4.8 operator”!=() [2/2]	92
11.1.4.9 operator<<() [1/59]	92
11.1.4.10 operator<<() [2/59]	93
11.1.4.11 operator<<() [3/59]	93
11.1.4.12 operator<<() [4/59]	93
11.1.4.13 operator<<() [5/59]	93
11.1.4.14 operator<<() [6/59]	93
11.1.4.15 operator<<() [7/59]	93

11.1.4.16 operator<<() [8/59]	93
11.1.4.17 operator<<() [9/59]	94
11.1.4.18 operator<<() [10/59]	94
11.1.4.19 operator<<() [11/59]	94
11.1.4.20 operator<<() [12/59]	94
11.1.4.21 operator<<() [13/59]	94
11.1.4.22 operator<<() [14/59]	94
11.1.4.23 operator<<() [15/59]	94
11.1.4.24 operator<<() [16/59]	95
11.1.4.25 operator<<() [17/59]	95
11.1.4.26 operator<<() [18/59]	95
11.1.4.27 operator<<() [19/59]	95
11.1.4.28 operator<<() [20/59]	95
11.1.4.29 operator<<() [21/59]	95
11.1.4.30 operator<<() [22/59]	95
11.1.4.31 operator<<() [23/59]	96
11.1.4.32 operator<<() [24/59]	96
11.1.4.33 operator<<() [25/59]	96
11.1.4.34 operator<<() [26/59]	96
11.1.4.35 operator<<() [27/59]	96
11.1.4.36 operator<<() [28/59]	96
11.1.4.37 operator<<() [29/59]	96
11.1.4.38 operator<<() [30/59]	97
11.1.4.39 operator<<() [31/59]	97
11.1.4.40 operator<<() [32/59]	97
11.1.4.41 operator<<() [33/59]	97
11.1.4.42 operator<<() [34/59]	97
11.1.4.43 operator<<() [35/59]	97
11.1.4.44 operator<<() [36/59]	97
11.1.4.45 operator<<() [37/59]	98
11.1.4.46 operator<<() [38/59]	98
11.1.4.47 operator<<() [39/59]	98
11.1.4.48 operator<<() [40/59]	98
11.1.4.49 operator<<() [41/59]	98
11.1.4.50 operator<<() [42/59]	98
11.1.4.51 operator<<() [43/59]	98
11.1.4.52 operator<<() [44/59]	99
11.1.4.53 operator<<() [45/59]	99
11.1.4.54 operator<<() [46/59]	99
11.1.4.55 operator<<() [47/59]	99

11.1.4.56	operator<<()	[48/59]	99
11.1.4.57	operator<<()	[49/59]	99
11.1.4.58	operator<<()	[50/59]	99
11.1.4.59	operator<<()	[51/59]	100
11.1.4.60	operator<<()	[52/59]	100
11.1.4.61	operator<<()	[53/59]	100
11.1.4.62	operator<<()	[54/59]	100
11.1.4.63	operator<<()	[55/59]	100
11.1.4.64	operator<<()	[56/59]	100
11.1.4.65	operator<<()	[57/59]	100
11.1.4.66	operator<<()	[58/59]	101
11.1.4.67	operator<<()	[59/59]	101
11.1.4.68	operator==()		101
11.1.4.69	operator>>()	[1/3]	101
11.1.4.70	operator>>()	[2/3]	101
11.1.4.71	operator>>()	[3/3]	101
11.1.4.72	Round()		101
11.1.4.73	roundat()		102
11.1.4.74	x16printf()		102
11.1.5	Variable Documentation		102
11.1.5.1	GlobalInstance		102
11.2	gdcm::network Namespace Reference		102
11.2.1	Enumeration Type Documentation		107
11.2.1.1	EEventID		107
11.2.1.2	EStateID		107
11.2.2	Function Documentation		108
11.2.2.1	GetStateIndex()		108
11.2.3	Variable Documentation		108
11.2.3.1	cMaxEventID		108
11.2.3.2	cMaxStateID		108
11.3	gdcm::SegmentHelper Namespace Reference		108
11.4	gdcm::terminal Namespace Reference		109
11.4.1	Detailed Description		109
11.4.2	Enumeration Type Documentation		110
11.4.2.1	Attribute		110
11.4.2.2	Color		110
11.4.2.3	Mode		110
11.4.3	Function Documentation		111
11.4.3.1	setAttribute()		111
11.4.3.2	setbgcolor()		111

11.4.3.3	setfgcolor()	111
11.4.3.4	setmode()	111
12	Class Documentation	113
12.1	gdcn::network::AAbortPDU Class Reference	113
12.1.1	Detailed Description	114
12.1.2	Constructor & Destructor Documentation	114
12.1.2.1	AAbortPDU()	114
12.1.3	Member Function Documentation	114
12.1.3.1	IsLastFragment()	114
12.1.3.2	Print()	114
12.1.3.3	Read()	115
12.1.3.4	SetReason()	115
12.1.3.5	SetSource()	115
12.1.3.6	Size()	115
12.1.3.7	Write()	115
12.2	gdcn::network::AAssociateACPDU Class Reference	116
12.2.1	Detailed Description	117
12.2.2	Member Typedef Documentation	117
12.2.2.1	SizeType	117
12.2.3	Constructor & Destructor Documentation	117
12.2.3.1	AAssociateACPDU()	117
12.2.4	Member Function Documentation	118
12.2.4.1	AddPresentationContextAC()	118
12.2.4.2	GetNumberOfPresentationContextAC()	118
12.2.4.3	GetPresentationContextAC()	118
12.2.4.4	GetUserInformation()	118
12.2.4.5	InitFromRQ()	118
12.2.4.6	IsLastFragment()	118
12.2.4.7	Print()	118
12.2.4.8	Read()	119
12.2.4.9	SetCalledAETitle()	119
12.2.4.10	SetCallingAETitle()	119
12.2.4.11	Size()	119
12.2.4.12	Write()	119
12.2.5	Friends And Related Symbol Documentation	119
12.2.5.1	AAssociateRQPDU	119
12.3	gdcn::network::AAssociateRJPDU Class Reference	120
12.3.1	Detailed Description	121
12.3.2	Constructor & Destructor Documentation	121

12.3.2.1	AAssociateRJPDU()	121
12.3.3	Member Function Documentation	121
12.3.3.1	IsLastFragment()	121
12.3.3.2	Print()	121
12.3.3.3	Read()	121
12.3.3.4	Size()	121
12.3.3.5	Write()	122
12.4	gdcm::network::AAssociateRQPDU Class Reference	122
12.4.1	Detailed Description	124
12.4.2	Member Typedef Documentation	124
12.4.2.1	PresentationContextArrayType	124
12.4.2.2	SizeType	124
12.4.3	Constructor & Destructor Documentation	124
12.4.3.1	AAssociateRQPDU() [1/2]	124
12.4.3.2	AAssociateRQPDU() [2/2]	124
12.4.4	Member Function Documentation	124
12.4.4.1	AddPresentationContext()	124
12.4.4.2	GetCalledAETitle()	124
12.4.4.3	GetCallingAETitle()	125
12.4.4.4	GetNumberOfPresentationContext()	125
12.4.4.5	GetPresentationContext()	125
12.4.4.6	GetPresentationContextByAbstractSyntax()	125
12.4.4.7	GetPresentationContextByID()	125
12.4.4.8	GetPresentationContexts()	125
12.4.4.9	GetReserved43_74()	125
12.4.4.10	GetUserInfoation()	125
12.4.4.11	IsAETitleValid()	126
12.4.4.12	IsLastFragment()	126
12.4.4.13	Print()	126
12.4.4.14	Read()	126
12.4.4.15	SetCalledAETitle()	126
12.4.4.16	SetCallingAETitle()	126
12.4.4.17	SetUserInfoation()	127
12.4.4.18	Size()	127
12.4.4.19	Write()	127
12.4.5	Friends And Related Symbol Documentation	127
12.4.5.1	AAssociateACPDU	127
12.5	gdcm::AbortEvent Class Reference	128
12.6	gdcm::network::AbstractSyntax Class Reference	129
12.6.1	Detailed Description	129

12.6.2 Constructor & Destructor Documentation	129
12.6.2.1 AbstractSyntax()	129
12.6.3 Member Function Documentation	130
12.6.3.1 GetAsDataElement()	130
12.6.3.2 GetName()	130
12.6.3.3 operator==(())	130
12.6.3.4 Print()	130
12.6.3.5 Read()	130
12.6.3.6 SetName()	130
12.6.3.7 SetNameFromUID()	130
12.6.3.8 Size()	130
12.6.3.9 Write()	131
12.7 gdcm::AnonymizeEvent Class Reference	131
12.7.1 Detailed Description	133
12.7.2 Member Typedef Documentation	133
12.7.2.1 Self	133
12.7.2.2 Superclass	133
12.7.3 Constructor & Destructor Documentation	133
12.7.3.1 AnonymizeEvent() [1/2]	133
12.7.3.2 ~AnonymizeEvent()	133
12.7.3.3 AnonymizeEvent() [2/2]	133
12.7.4 Member Function Documentation	134
12.7.4.1 CheckEvent()	134
12.7.4.2 GetEventName()	134
12.7.4.3 GetTag()	134
12.7.4.4 MakeObject()	134
12.7.4.5 operator=()	134
12.7.4.6 SetTag()	134
12.8 gdcm::Anonymizer Class Reference	135
12.8.1 Detailed Description	137
12.8.2 Constructor & Destructor Documentation	138
12.8.2.1 Anonymizer()	138
12.8.2.2 ~Anonymizer()	139
12.8.3 Member Function Documentation	139
12.8.3.1 BALCPPProtect()	139
12.8.3.2 BasicApplicationLevelConfidentialityProfile()	139
12.8.3.3 CanEmptyTag()	139
12.8.3.4 Clear() [1/2]	139
12.8.3.5 Clear() [2/2]	139
12.8.3.6 ClearInternalUIDs()	140

12.8.3.7 Empty() [1/2]	140
12.8.3.8 Empty() [2/2]	140
12.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()	140
12.8.3.10 GetCryptographicMessageSyntax()	140
12.8.3.11 GetFile()	141
12.8.3.12 New()	141
12.8.3.13 RecurseDataSet()	141
12.8.3.14 Remove() [1/2]	141
12.8.3.15 Remove() [2/2]	141
12.8.3.16 RemoveGroupLength()	141
12.8.3.17 RemovePrivateTags()	142
12.8.3.18 RemoveRetired()	142
12.8.3.19 Replace() [1/4]	142
12.8.3.20 Replace() [2/4]	142
12.8.3.21 Replace() [3/4]	142
12.8.3.22 Replace() [4/4]	143
12.8.3.23 SetCryptographicMessageSyntax()	143
12.8.3.24 SetFile()	143
12.9 gdcmm::AnyEvent Class Reference	144
12.10 gdcmm::network::ApplicationContext Class Reference	145
12.10.1 Detailed Description	146
12.10.2 Constructor & Destructor Documentation	146
12.10.2.1 ApplicationContext()	146
12.10.3 Member Function Documentation	146
12.10.3.1 GetName()	146
12.10.3.2 Print()	146
12.10.3.3 Read()	146
12.10.3.4 SetName()	146
12.10.3.5 Size()	146
12.10.3.6 Write()	147
12.11 gdcmm::ApplicationEntity Class Reference	147
12.11.1 Detailed Description	148
12.11.2 Member Function Documentation	148
12.11.2.1 IsValid()	148
12.11.2.2 Print()	148
12.11.2.3 SetBlob()	148
12.11.2.4 Squeeze()	148
12.11.3 Member Data Documentation	149
12.11.3.1 Internal	149
12.11.3.2 MaxLength	149

12.11.3.3 MaxNumberOfComponents	149
12.11.3.4 Padding	149
12.11.3.5 Separator	149
12.12 gdcmm::network::AReleaseRPPDU Class Reference	149
12.12.1 Detailed Description	150
12.12.2 Constructor & Destructor Documentation	150
12.12.2.1 AReleaseRPPDU()	150
12.12.3 Member Function Documentation	150
12.12.3.1 IsLastFragment()	150
12.12.3.2 Print()	151
12.12.3.3 Read()	151
12.12.3.4 Size()	151
12.12.3.5 Write()	151
12.13 gdcmm::network::AReleaseRQPDU Class Reference	151
12.13.1 Detailed Description	152
12.13.2 Constructor & Destructor Documentation	153
12.13.2.1 AReleaseRQPDU()	153
12.13.3 Member Function Documentation	153
12.13.3.1 IsLastFragment()	153
12.13.3.2 Print()	153
12.13.3.3 Read()	153
12.13.3.4 Size()	153
12.13.3.5 Write()	153
12.14 gdcmm::network::ARTIMTimer Class Reference	154
12.14.1 Detailed Description	154
12.14.2 Constructor & Destructor Documentation	154
12.14.2.1 ARTIMTimer()	154
12.14.3 Member Function Documentation	154
12.14.3.1 GetElapsedTime()	154
12.14.3.2 GetHasExpired()	155
12.14.3.3 GetTimeout()	155
12.14.3.4 SetTimeout()	155
12.14.3.5 Start()	155
12.14.3.6 Stop()	155
12.15 gdcmm::ASN1 Class Reference	155
12.15.1 Detailed Description	156
12.15.2 Constructor & Destructor Documentation	156
12.15.2.1 ASN1() [1/2]	156
12.15.2.2 ~ASN1()	156
12.15.2.3 ASN1() [2/2]	156

12.15.3 Member Function Documentation	156
12.15.3.1 operator=()	156
12.15.3.2 ParseDump()	156
12.15.3.3 ParseDumpFile()	157
12.15.3.4 TestPBKDF2()	157
12.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference	157
12.16.1 Detailed Description	157
12.16.2 Constructor & Destructor Documentation	157
12.16.2.1 AsynchronousOperationsWindowSub()	157
12.16.3 Member Function Documentation	158
12.16.3.1 Print()	158
12.16.3.2 Read()	158
12.16.3.3 Size()	158
12.16.3.4 Write()	158
12.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference	158
12.17.1 Detailed Description	160
12.17.2 Member Typedef Documentation	161
12.17.2.1 ArrayType	161
12.17.3 Member Enumeration Documentation	161
12.17.3.1 anonymous enum	161
12.17.4 Member Function Documentation	161
12.17.4.1 GDCM_STATIC_ASSERT() [1/3]	161
12.17.4.2 GDCM_STATIC_ASSERT() [2/3]	161
12.17.4.3 GDCM_STATIC_ASSERT() [3/3]	162
12.17.4.4 GetAsDataElement()	162
12.17.4.5 GetDictVM()	162
12.17.4.6 GetDictVR()	162
12.17.4.7 GetNumberOfValues()	162
12.17.4.8 GetTag()	163
12.17.4.9 GetValue() [1/2]	163
12.17.4.10 GetValue() [2/2]	163
12.17.4.11 GetValues()	163
12.17.4.12 GetVM()	164
12.17.4.13 GetVR()	164
12.17.4.14 operator"! =()	164
12.17.4.15 operator<()	164
12.17.4.16 operator==()	164
12.17.4.17 operator[]() [1/2]	165
12.17.4.18 operator[]() [2/2]	165
12.17.4.19 Print()	165

12.17.4.20	Set()	165
12.17.4.21	SetByteValue()	165
12.17.4.22	SetByteValueNoSwap()	166
12.17.4.23	SetFromDataElement()	166
12.17.4.24	SetFromDataSet()	166
12.17.4.25	SetValue()	167
12.17.4.26	SetValues()	167
12.17.5	Member Data Documentation	167
12.17.5.1	Internal	167
12.18	gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	168
12.18.1	Member Typedef Documentation	170
12.18.1.1	ArrayType	170
12.18.2	Member Enumeration Documentation	171
12.18.2.1	anonymous enum	171
12.18.2.2	anonymous enum	171
12.18.3	Member Function Documentation	171
12.18.3.1	GDCM_STATIC_ASSERT() [1/4]	171
12.18.3.2	GDCM_STATIC_ASSERT() [2/4]	171
12.18.3.3	GDCM_STATIC_ASSERT() [3/4]	171
12.18.3.4	GDCM_STATIC_ASSERT() [4/4]	171
12.18.3.5	GetAsDataElement()	172
12.18.3.6	GetDictVM()	172
12.18.3.7	GetDictVR()	172
12.18.3.8	GetNumberOfValues()	172
12.18.3.9	GetTag()	172
12.18.3.10	GetValue() [1/2]	172
12.18.3.11	GetValue() [2/2]	172
12.18.3.12	GetValues()	173
12.18.3.13	GetVM()	173
12.18.3.14	GetVR()	173
12.18.3.15	operator"!=()	173
12.18.3.16	operator<()	173
12.18.3.17	operator==()	173
12.18.3.18	operator[]()	174
12.18.3.19	Print()	174
12.18.3.20	Set()	174
12.18.3.21	SetByteValue()	174
12.18.3.22	SetByteValueNoSwap()	174
12.18.3.23	SetFromDataElement()	175
12.18.3.24	SetFromDataSet()	175

12.18.3.25 SetValue()	175
12.18.3.26 SetValues()	175
12.18.4 Member Data Documentation	175
12.18.4.1 Internal	175
12.19 gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	176
12.19.1 Member Typedef Documentation	178
12.19.1.1 ArrayType	178
12.19.2 Member Enumeration Documentation	178
12.19.2.1 anonymous enum	178
12.19.3 Member Function Documentation	178
12.19.3.1 GDCM_STATIC_ASSERT()	178
12.19.3.2 GetAsDataElement()	178
12.19.3.3 GetDictVM()	178
12.19.3.4 GetDictVR()	178
12.19.3.5 GetNumberOfValues()	179
12.19.3.6 GetTag()	179
12.19.3.7 GetValue()	179
12.19.3.8 GetValues()	179
12.19.3.9 GetVM()	179
12.19.3.10 GetVR()	179
12.19.3.11 operator"!=(179
12.19.3.12 operator<()	179
12.19.3.13 operator==(179
12.19.3.14 operator[]()	180
12.19.3.15 Print()	180
12.19.3.16 Set()	180
12.19.3.17 SetByteValue()	180
12.19.3.18 SetByteValueNoSwap()	180
12.19.3.19 SetFromDataElement()	180
12.19.3.20 SetFromDataSet()	180
12.19.3.21 SetValue()	180
12.19.3.22 SetValues()	181
12.19.4 Member Data Documentation	181
12.19.4.1 Internal	181
12.20 gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	181
12.20.1 Member Typedef Documentation	183
12.20.1.1 ArrayType	183
12.20.2 Member Enumeration Documentation	183
12.20.2.1 anonymous enum	183
12.20.3 Member Function Documentation	184

12.20.3.1	GDCM_STATIC_ASSERT()	184
12.20.3.2	GetAsDataElement()	184
12.20.3.3	GetDictVM()	184
12.20.3.4	GetDictVR()	184
12.20.3.5	GetNumberOfValues()	184
12.20.3.6	GetTag()	184
12.20.3.7	GetValue()	184
12.20.3.8	GetValues()	184
12.20.3.9	GetVM()	184
12.20.3.10	GetVR()	185
12.20.3.11	operator"!=()	185
12.20.3.12	operator<()	185
12.20.3.13	operator==()	185
12.20.3.14	operator[]()	185
12.20.3.15	Print()	185
12.20.3.16	Set()	185
12.20.3.17	SetByteValue()	185
12.20.3.18	SetByteValueNoSwap()	186
12.20.3.19	SetFromDataElement()	186
12.20.3.20	SetFromDataSet()	186
12.20.3.21	SetValue()	186
12.20.3.22	SetValues()	186
12.20.4	Member Data Documentation	186
12.20.4.1	Internal	186
12.21	gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	187
12.21.1	Member Typedef Documentation	189
12.21.1.1	ArrayType	189
12.21.2	Member Enumeration Documentation	189
12.21.2.1	anonymous enum	189
12.21.3	Constructor & Destructor Documentation	189
12.21.3.1	Attribute()	189
12.21.3.2	~Attribute()	189
12.21.4	Member Function Documentation	189
12.21.4.1	GDCM_STATIC_ASSERT() [1/3]	189
12.21.4.2	GDCM_STATIC_ASSERT() [2/3]	189
12.21.4.3	GDCM_STATIC_ASSERT() [3/3]	190
12.21.4.4	GetAsDataElement()	190
12.21.4.5	GetDictVM()	190
12.21.4.6	GetDictVR()	190
12.21.4.7	GetNumberOfValues()	190

12.21.4.8	GetTag()	190
12.21.4.9	GetValue() [1/2]	191
12.21.4.10	GetValue() [2/2]	191
12.21.4.11	GetValues()	191
12.21.4.12	GetVM()	191
12.21.4.13	GetVR()	191
12.21.4.14	operator"!=()	191
12.21.4.15	operator<()	192
12.21.4.16	operator==()	192
12.21.4.17	operator[]() [1/2]	192
12.21.4.18	operator[]() [2/2]	192
12.21.4.19	Print()	192
12.21.4.20	Set()	192
12.21.4.21	SetByteValue()	193
12.21.4.22	SetByteValueNoSwap()	193
12.21.4.23	SetFromDataElement()	193
12.21.4.24	SetFromDataSet()	193
12.21.4.25	SetNumberOfValues()	193
12.21.4.26	SetValue() [1/2]	194
12.21.4.27	SetValue() [2/2]	194
12.21.4.28	SetValues()	194
12.22	gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference	194
12.22.1	Member Typedef Documentation	198
12.22.1.1	ArrayType	198
12.22.2	Member Enumeration Documentation	198
12.22.2.1	anonymous enum	198
12.22.3	Member Function Documentation	198
12.22.3.1	GDCM_STATIC_ASSERT()	198
12.22.3.2	GetAsDataElement()	198
12.22.3.3	GetDictVM()	198
12.22.3.4	GetDictVR()	198
12.22.3.5	GetNumberOfValues()	198
12.22.3.6	GetTag()	199
12.22.3.7	GetValue()	199
12.22.3.8	GetValues()	199
12.22.3.9	GetVM()	199
12.22.3.10	GetVR()	199
12.22.3.11	operator"!=()	199
12.22.3.12	operator<()	199
12.22.3.13	operator==()	199

12.22.3.14	operator[]()	200
12.22.3.15	Print()	200
12.22.3.16	Set()	200
12.22.3.17	SetByteValue()	200
12.22.3.18	SetByteValueNoSwap()	200
12.22.3.19	SetFromDataElement()	200
12.22.3.20	SetFromDataSet()	200
12.22.3.21	SetValue()	200
12.22.3.22	SetValues()	201
12.22.4	Member Data Documentation	201
12.22.4.1	Internal	201
12.23	gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	201
12.23.1	Member Typedef Documentation	203
12.23.1.1	ArrayType	203
12.23.2	Member Enumeration Documentation	203
12.23.2.1	anonymous enum	203
12.23.3	Member Function Documentation	204
12.23.3.1	GDCM_STATIC_ASSERT()	204
12.23.3.2	GetAsDataElement()	204
12.23.3.3	GetDictVM()	204
12.23.3.4	GetDictVR()	204
12.23.3.5	GetNumberOfValues()	204
12.23.3.6	GetTag()	204
12.23.3.7	GetValue()	204
12.23.3.8	GetValues()	204
12.23.3.9	GetVM()	204
12.23.3.10	GetVR()	205
12.23.3.11	operator"!=()	205
12.23.3.12	operator<()	205
12.23.3.13	operator==(())	205
12.23.3.14	operator[]()	205
12.23.3.15	Print()	205
12.23.3.16	Set()	205
12.23.3.17	SetByteValue()	205
12.23.3.18	SetByteValueNoSwap()	206
12.23.3.19	SetFromDataElement()	206
12.23.3.20	SetFromDataSet()	206
12.23.3.21	SetValue()	206
12.23.3.22	SetValues()	206
12.23.4	Member Data Documentation	206

12.23.4.1 Internal	206
12.24 gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	207
12.24.1 Member Typedef Documentation	210
12.24.1.1 ArrayType	210
12.24.2 Member Enumeration Documentation	210
12.24.2.1 anonymous enum	210
12.24.3 Member Function Documentation	210
12.24.3.1 GDCM_STATIC_ASSERT()	210
12.24.3.2 GetAsDataElement()	211
12.24.3.3 GetDictVM()	211
12.24.3.4 GetDictVR()	211
12.24.3.5 GetNumberOfValues()	211
12.24.3.6 GetTag()	211
12.24.3.7 GetValue()	211
12.24.3.8 GetValues()	211
12.24.3.9 GetVM()	211
12.24.3.10 GetVR()	211
12.24.3.11 operator"!=()	212
12.24.3.12 operator<()	212
12.24.3.13 operator==()	212
12.24.3.14 operator[]()	212
12.24.3.15 Print()	212
12.24.3.16 Set()	212
12.24.3.17 SetByteValue()	212
12.24.3.18 SetByteValueNoSwap()	212
12.24.3.19 SetFromDataElement()	213
12.24.3.20 SetFromDataSet()	213
12.24.3.21 SetValue()	213
12.24.3.22 SetValues()	213
12.24.4 Member Data Documentation	213
12.24.4.1 Internal	213
12.25 gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	214
12.25.1 Member Typedef Documentation	216
12.25.1.1 ArrayType	216
12.25.2 Member Enumeration Documentation	216
12.25.2.1 anonymous enum	216
12.25.3 Member Function Documentation	216
12.25.3.1 GDCM_STATIC_ASSERT()	216
12.25.3.2 GetAsDataElement()	216
12.25.3.3 GetDictVM()	216

12.25.3.4	GetDictVR()	216
12.25.3.5	GetNumberOfValues()	217
12.25.3.6	GetTag()	217
12.25.3.7	GetValue()	217
12.25.3.8	GetValues()	217
12.25.3.9	GetVM()	217
12.25.3.10	GetVR()	217
12.25.3.11	operator"!=()	217
12.25.3.12	operator<()	217
12.25.3.13	operator==(())	217
12.25.3.14	operator[]()	218
12.25.3.15	Print()	218
12.25.3.16	Set()	218
12.25.3.17	SetByteValue()	218
12.25.3.18	SetByteValueNoSwap()	218
12.25.3.19	SetFromDataElement()	218
12.25.3.20	SetFromDataSet()	218
12.25.3.21	SetValue()	218
12.25.3.22	SetValues()	219
12.25.4	Member Data Documentation	219
12.25.4.1	Internal	219
12.26	gdcm::AudioCodec Class Reference	219
12.26.1	Detailed Description	221
12.26.2	Constructor & Destructor Documentation	221
12.26.2.1	AudioCodec()	221
12.26.2.2	~AudioCodec()	221
12.26.3	Member Function Documentation	221
12.26.3.1	CanCode()	221
12.26.3.2	CanDecode()	221
12.26.3.3	Decode()	222
12.27	gdcm::Base64 Class Reference	222
12.27.1	Detailed Description	222
12.27.2	Constructor & Destructor Documentation	222
12.27.2.1	Base64()	222
12.27.3	Member Function Documentation	223
12.27.3.1	Decode()	223
12.27.3.2	Encode()	223
12.27.3.3	GetDecodeLength()	224
12.27.3.4	GetEncodeLength()	224
12.27.3.5	operator==(())	224

12.28	gdcm::network::BaseCompositeMessage Class Reference	224
12.28.1	Detailed Description	225
12.28.2	Constructor & Destructor Documentation	226
12.28.2.1	~BaseCompositeMessage()	226
12.28.3	Member Function Documentation	226
12.28.3.1	ConstructPDV()	226
12.29	gdcm::network::BaseNormalizedMessage Class Reference	226
12.29.1	Detailed Description	227
12.29.2	Constructor & Destructor Documentation	228
12.29.2.1	~BaseNormalizedMessage()	228
12.29.3	Member Function Documentation	228
12.29.3.1	ConstructPDV()	228
12.30	gdcm::network::BasePDU Class Reference	229
12.30.1	Detailed Description	229
12.30.2	Constructor & Destructor Documentation	230
12.30.2.1	~BasePDU()	230
12.30.3	Member Function Documentation	230
12.30.3.1	IsLastFragment()	230
12.30.3.2	Print()	230
12.30.3.3	Read()	230
12.30.3.4	Size()	231
12.30.3.5	Write()	231
12.31	gdcm::BaseQuery Class Reference	231
12.31.1	Detailed Description	233
12.31.2	Constructor & Destructor Documentation	233
12.31.2.1	BaseQuery()	233
12.31.2.2	~BaseQuery()	233
12.31.3	Member Function Documentation	233
12.31.3.1	AddQueryDataSet()	233
12.31.3.2	GetAbstractSyntaxUID()	234
12.31.3.3	GetQueryDataSet() [1/2]	234
12.31.3.4	GetQueryDataSet() [2/2]	234
12.31.3.5	GetSOPInstanceUID()	234
12.31.3.6	Print()	234
12.31.3.7	SetSearchParameter() [1/3]	234
12.31.3.8	SetSearchParameter() [2/3]	234
12.31.3.9	SetSearchParameter() [3/3]	235
12.31.3.10	SetSOPInstanceUID()	235
12.31.3.11	ValidateQuery()	235
12.31.3.12	ValidDataSet()	235

12.31.3.13 WriteHelpFile()	235
12.31.3.14 WriteQuery()	235
12.31.4 Friends And Related Symbol Documentation	235
12.31.4.1 QueryFactory	235
12.31.5 Member Data Documentation	236
12.31.5.1 mDataSet	236
12.31.5.2 mSopInstanceUID	236
12.32 gdcmm::BaseRootQuery Class Reference	236
12.32.1 Detailed Description	238
12.32.2 Constructor & Destructor Documentation	238
12.32.2.1 BaseRootQuery()	238
12.32.2.2 ~BaseRootQuery()	239
12.32.3 Member Function Documentation	239
12.32.3.1 Construct()	239
12.32.3.2 GetQueryLevelFromQueryRoot()	239
12.32.3.3 GetQueryLevelFromString()	239
12.32.3.4 GetQueryLevelString()	239
12.32.3.5 GetTagListByLevel()	239
12.32.3.6 InitializeDataSet()	239
12.32.3.7 ValidateQuery()	240
12.32.4 Friends And Related Symbol Documentation	240
12.32.4.1 QueryFactory	240
12.32.5 Member Data Documentation	240
12.32.5.1 mHelpDescription	240
12.32.5.2 mImage	240
12.32.5.3 mPatient	240
12.32.5.4 mRootType	241
12.32.5.5 mSeries	241
12.32.5.6 mStudy	241
12.33 gdcmm::SegmentHelper::BasicCodedEntry Struct Reference	241
12.33.1 Detailed Description	242
12.33.2 Constructor & Destructor Documentation	242
12.33.2.1 BasicCodedEntry() [1/3]	242
12.33.2.2 BasicCodedEntry() [2/3]	242
12.33.2.3 BasicCodedEntry() [3/3]	243
12.33.3 Member Function Documentation	243
12.33.3.1 IsEmpty()	243
12.33.4 Member Data Documentation	243
12.33.4.1 CM	243
12.33.4.2 CSD	243

12.33.4.3 CSV	244
12.33.4.4 CV	244
12.34 gdcm::BasicOffsetTable Class Reference	244
12.34.1 Detailed Description	247
12.34.2 Constructor & Destructor Documentation	247
12.34.2.1 BasicOffsetTable()	247
12.34.3 Member Function Documentation	247
12.34.3.1 Read()	247
12.34.4 Friends And Related Symbol Documentation	248
12.34.4.1 operator<<	248
12.35 gdcm::Bitmap Class Reference	248
12.35.1 Detailed Description	251
12.35.2 Member Typedef Documentation	251
12.35.2.1 LUTPtr	251
12.35.3 Constructor & Destructor Documentation	251
12.35.3.1 Bitmap()	251
12.35.3.2 ~Bitmap()	252
12.35.4 Member Function Documentation	252
12.35.4.1 AreOverlaysInPixelData()	252
12.35.4.2 Clear()	252
12.35.4.3 ComputeLossyFlag()	252
12.35.4.4 GetBuffer()	252
12.35.4.5 GetBuffer2()	252
12.35.4.6 GetBufferLength()	253
12.35.4.7 GetColumns()	253
12.35.4.8 GetDataElement() [1/2]	253
12.35.4.9 GetDataElement() [2/2]	253
12.35.4.10 GetDimension()	253
12.35.4.11 GetDimensions()	254
12.35.4.12 GetLUT() [1/2]	254
12.35.4.13 GetLUT() [2/2]	254
12.35.4.14 GetNeedByteSwap()	254
12.35.4.15 GetNumberOfDimensions()	254
12.35.4.16 GetPhotometricInterpretation()	255
12.35.4.17 GetPixelFormat() [1/2]	255
12.35.4.18 GetPixelFormat() [2/2]	255
12.35.4.19 GetPlanarConfiguration()	255
12.35.4.20 GetRows()	255
12.35.4.21 GetTransferSyntax()	256
12.35.4.22 IsEmpty()	256

12.35.4.23	IsLossy()	256
12.35.4.24	IsTransferSyntaxCompatible()	256
12.35.4.25	Print()	256
12.35.4.26	SetColumns()	256
12.35.4.27	SetDataElement()	257
12.35.4.28	SetDimension()	257
12.35.4.29	SetDimensions()	257
12.35.4.30	SetLossyFlag()	257
12.35.4.31	SetLUT()	258
12.35.4.32	SetNeedByteSwap()	258
12.35.4.33	SetNumberOfDimensions()	258
12.35.4.34	SetPhotometricInterpretation()	258
12.35.4.35	SetPixelFormat()	258
12.35.4.36	SetPlanarConfiguration()	259
12.35.4.37	SetRows()	259
12.35.4.38	SetTransferSyntax()	259
12.35.4.39	TryJPEG2000Codec()	259
12.35.4.40	TryJPEG2000Codec2()	259
12.35.4.41	TryJPEGCodec()	259
12.35.4.42	TryJPEGCodec2()	260
12.35.4.43	TryJPEGLSCodec()	260
12.35.4.44	TryKAKADUCodec()	260
12.35.4.45	TryPVRGCodec()	260
12.35.4.46	TryRAWCodec()	260
12.35.4.47	TryRLECodec()	260
12.35.4.48	UnusedBitsPresentInPixelData()	260
12.35.5	Friends And Related Symbol Documentation	261
12.35.5.1	ImageChangeTransferSyntax	261
12.35.5.2	PixmapReader	261
12.35.6	Member Data Documentation	261
12.35.6.1	Dimensions	261
12.35.6.2	LossyFlag	261
12.35.6.3	LUT	261
12.35.6.4	NeedByteSwap	261
12.35.6.5	NumberOfDimensions	262
12.35.6.6	PF	262
12.35.6.7	PI	262
12.35.6.8	PixelData	262
12.35.6.9	PlanarConfiguration	262
12.35.6.10	TS	262

12.36	gdcm::BitmapToBitmapFilter Class Reference	263
12.36.1	Detailed Description	264
12.36.2	Constructor & Destructor Documentation	264
12.36.2.1	BitmapToBitmapFilter()	264
12.36.2.2	~BitmapToBitmapFilter()	264
12.36.3	Member Function Documentation	264
12.36.3.1	GetOutput()	264
12.36.3.2	GetOutputAsBitmap()	264
12.36.3.3	SetInput()	265
12.36.4	Member Data Documentation	265
12.36.4.1	Input	265
12.36.4.2	Output	265
12.37	gdcm::BoxRegion Class Reference	265
12.37.1	Detailed Description	267
12.37.2	Constructor & Destructor Documentation	267
12.37.2.1	BoxRegion() [1/2]	267
12.37.2.2	~BoxRegion()	267
12.37.2.3	BoxRegion() [2/2]	267
12.37.3	Member Function Documentation	267
12.37.3.1	Area()	267
12.37.3.2	BoundingBox()	268
12.37.3.3	Clone()	268
12.37.3.4	ComputeBoundingBox()	268
12.37.3.5	Empty()	268
12.37.3.6	GetXMax()	268
12.37.3.7	GetXMin()	268
12.37.3.8	GetYMax()	269
12.37.3.9	GetYMin()	269
12.37.3.10	GetZMax()	269
12.37.3.11	GetZMin()	269
12.37.3.12	IsValid()	269
12.37.3.13	operator=()	269
12.37.3.14	Print()	269
12.37.3.15	SetDomain()	270
12.38	gdcm::ByteBuffer Class Reference	270
12.38.1	Detailed Description	270
12.38.2	Constructor & Destructor Documentation	271
12.38.2.1	ByteBuffer()	271
12.38.3	Member Function Documentation	271
12.38.3.1	Get()	271

12.38.3.2	GetStart()	271
12.38.3.3	ShiftEnd()	271
12.38.3.4	UpdatePosition()	271
12.39	gdcmm::ByteSwap< T > Class Template Reference	271
12.39.1	Detailed Description	272
12.39.2	Member Function Documentation	272
12.39.2.1	Swap()	272
12.39.2.2	SwapFromSwapCodeIntoSystem()	272
12.39.2.3	SwapRange()	272
12.39.2.4	SwapRangeFromSwapCodeIntoSystem()	273
12.39.2.5	SystemIsBigEndian()	273
12.39.2.6	SystemIsLittleEndian()	273
12.40	gdcmm::ByteSwapFilter Class Reference	273
12.40.1	Detailed Description	274
12.40.2	Constructor & Destructor Documentation	274
12.40.2.1	ByteSwapFilter() [1/2]	274
12.40.2.2	~ByteSwapFilter()	274
12.40.2.3	ByteSwapFilter() [2/2]	274
12.40.3	Member Function Documentation	274
12.40.3.1	ByteSwap()	274
12.40.3.2	operator=()	275
12.40.3.3	SetByteSwapTag()	275
12.41	gdcmm::ByteValue Class Reference	275
12.41.1	Detailed Description	277
12.41.2	Constructor & Destructor Documentation	278
12.41.2.1	ByteValue() [1/2]	278
12.41.2.2	ByteValue() [2/2]	278
12.41.2.3	~ByteValue()	278
12.41.3	Member Function Documentation	278
12.41.3.1	Append()	278
12.41.3.2	Clear()	278
12.41.3.3	ComputeLength()	278
12.41.3.4	Fill()	279
12.41.3.5	GetBuffer()	279
12.41.3.6	GetLength()	279
12.41.3.7	GetPointer()	280
12.41.3.8	GetVoidPointer() [1/2]	280
12.41.3.9	GetVoidPointer() [2/2]	280
12.41.3.10	IsEmpty()	280
12.41.3.11	IsPrintable()	280

12.41.3.12 operator const std::vector< char > &()	281
12.41.3.13 operator=()	281
12.41.3.14 operator==() [1/2]	281
12.41.3.15 operator==() [2/2]	281
12.41.3.16 Print()	281
12.41.3.17 PrintASCII()	281
12.41.3.18 PrintASCIIXML()	282
12.41.3.19 PrintGroupLength()	282
12.41.3.20 PrintHex()	282
12.41.3.21 PrintHexXML()	282
12.41.3.22 PrintPNXML()	282
12.41.3.23 Read() [1/2]	282
12.41.3.24 Read() [2/2]	282
12.41.3.25 SetLength()	283
12.41.3.26 SetLengthOnly()	283
12.41.3.27 Write() [1/2]	283
12.41.3.28 Write() [2/2]	283
12.41.3.29 WriteBuffer()	283
12.42 gdcmm::CAPICryptoFactory Class Reference	284
12.42.1 Constructor & Destructor Documentation	285
12.42.1.1 CAPICryptoFactory()	285
12.42.2 Member Function Documentation	285
12.42.2.1 CreateCMSProvider()	285
12.43 gdcmm::CAPICryptographicMessageSyntax Class Reference	286
12.43.1 Constructor & Destructor Documentation	287
12.43.1.1 CAPICryptographicMessageSyntax()	287
12.43.1.2 ~CAPICryptographicMessageSyntax()	287
12.43.2 Member Function Documentation	287
12.43.2.1 Decrypt()	287
12.43.2.2 Encrypt()	288
12.43.2.3 GetCipherType()	288
12.43.2.4 GetInitialized()	288
12.43.2.5 ParseCertificateFile()	288
12.43.2.6 ParseKeyFile()	288
12.43.2.7 SetCipherType()	288
12.43.2.8 SetPassword()	289
12.44 gdcmm::network::CEchoRQ Class Reference	289
12.44.1 Detailed Description	290
12.44.2 Member Function Documentation	291
12.44.2.1 ConstructPDV()	291

12.44.3 Member Data Documentation	291
12.44.3.1 AffectedSOPClassUID	291
12.44.3.2 MessageID	291
12.45 gdcmm::network::CEchoRSP Class Reference	291
12.45.1 Detailed Description	292
12.45.2 Member Function Documentation	292
12.45.2.1 ConstructPDVByDataSet()	292
12.46 gdcmm::network::CFind Class Reference	292
12.46.1 Detailed Description	293
12.47 gdcmm::network::CFindCancelRQ Class Reference	293
12.47.1 Detailed Description	294
12.47.2 Member Function Documentation	294
12.47.2.1 ConstructPDVByDataSet()	294
12.48 gdcmm::network::CFindRQ Class Reference	294
12.48.1 Detailed Description	295
12.48.2 Member Function Documentation	295
12.48.2.1 ConstructPDV()	295
12.49 gdcmm::network::CFindRSP Class Reference	296
12.49.1 Detailed Description	297
12.49.2 Member Function Documentation	297
12.49.2.1 ConstructPDVByDataSet()	297
12.50 gdcmm::Cleaner Class Reference	297
12.50.1 Detailed Description	300
12.50.2 Member Typedef Documentation	300
12.50.2.1 CodedEntryData	300
12.50.3 Constructor & Destructor Documentation	300
12.50.3.1 Cleaner()	300
12.50.3.2 ~Cleaner()	300
12.50.4 Member Function Documentation	300
12.50.4.1 Clean()	300
12.50.4.2 Empty() [1/4]	301
12.50.4.3 Empty() [2/4]	301
12.50.4.4 Empty() [3/4]	301
12.50.4.5 Empty() [4/4]	301
12.50.4.6 EmptyWhenScrubFails()	301
12.50.4.7 GetFile()	301
12.50.4.8 New()	302
12.50.4.9 Preserve()	302
12.50.4.10 Remove() [1/4]	302
12.50.4.11 Remove() [2/4]	302

12.50.4.12 Remove() [3/4]	302
12.50.4.13 Remove() [4/4]	302
12.50.4.14 RemoveAllGroupLength()	303
12.50.4.15 RemoveAllIllegal()	303
12.50.4.16 RemoveAllMissingPrivateCreator()	303
12.50.4.17 RemoveMissingPrivateCreator()	303
12.50.4.18 ReplaceCodeMeaning()	303
12.50.4.19 Scrub() [1/4]	303
12.50.4.20 Scrub() [2/4]	303
12.50.4.21 Scrub() [3/4]	304
12.50.4.22 Scrub() [4/4]	304
12.50.4.23 SetFile()	304
12.51 gdcm::network::CMoveCancelRq Class Reference	304
12.51.1 Member Function Documentation	305
12.51.1.1 ConstructPDVByDataSet()	305
12.52 gdcm::network::CMoveRQ Class Reference	306
12.52.1 Detailed Description	307
12.52.2 Member Function Documentation	307
12.52.2.1 ConstructPDV()	307
12.53 gdcm::network::CMoveRSP Class Reference	307
12.53.1 Detailed Description	308
12.53.2 Member Function Documentation	308
12.53.2.1 ConstructPDVByDataSet()	308
12.54 gdcm::Codec Class Reference	309
12.54.1 Detailed Description	310
12.55 gdcm::Coder Class Reference	310
12.55.1 Detailed Description	311
12.55.2 Constructor & Destructor Documentation	311
12.55.2.1 ~Coder()	311
12.55.3 Member Function Documentation	311
12.55.3.1 CanCode()	311
12.55.3.2 Code()	311
12.55.3.3 InternalCode()	312
12.56 gdcm::CodeString Class Reference	312
12.56.1 Detailed Description	313
12.56.2 Member Typedef Documentation	313
12.56.2.1 const_iterator	313
12.56.2.2 const_reference	313
12.56.2.3 const_reverse_iterator	313
12.56.2.4 difference_type	314

12.56.2.5 iterator	314
12.56.2.6 pointer	314
12.56.2.7 reference	314
12.56.2.8 reverse_iterator	314
12.56.2.9 size_type	314
12.56.2.10 value_type	314
12.56.3 Constructor & Destructor Documentation	314
12.56.3.1 CodeString() [1/4]	314
12.56.3.2 CodeString() [2/4]	315
12.56.3.3 CodeString() [3/4]	315
12.56.3.4 CodeString() [4/4]	315
12.56.4 Member Function Documentation	315
12.56.4.1 GetAsString()	315
12.56.4.2 IsValid()	315
12.56.4.3 Size()	315
12.56.4.4 TrimInternal()	315
12.56.5 Friends And Related Symbol Documentation	316
12.56.5.1 operator"!="	316
12.56.5.2 operator<<	316
12.56.5.3 operator==	316
12.57 gdcmm::Command Class Reference	316
12.57.1 Detailed Description	318
12.57.2 Constructor & Destructor Documentation	318
12.57.2.1 Command() [1/2]	318
12.57.2.2 Command() [2/2]	318
12.57.2.3 ~Command()	318
12.57.3 Member Function Documentation	319
12.57.3.1 Execute() [1/2]	319
12.57.3.2 Execute() [2/2]	319
12.57.3.3 operator=()	319
12.58 gdcmm::CommandDataSet Class Reference	320
12.58.1 Detailed Description	322
12.58.2 Constructor & Destructor Documentation	322
12.58.2.1 CommandDataSet()	322
12.58.2.2 ~CommandDataSet()	323
12.58.3 Member Function Documentation	323
12.58.3.1 Insert()	323
12.58.3.2 Read()	323
12.58.3.3 Replace()	323
12.58.3.4 Write()	323

12.58.4 Friends And Related Symbol Documentation	324
12.58.4.1 operator<<	324
12.59 gdcmm::network::CompositeMessageFactory Class Reference	324
12.59.1 Detailed Description	324
12.59.2 Member Function Documentation	325
12.59.2.1 ConstructCEchoRQ()	325
12.59.2.2 ConstructCFindRQ()	325
12.59.2.3 ConstructCMoveRQ()	325
12.59.2.4 ConstructCStoreRQ()	325
12.59.2.5 ConstructCStoreRSP()	325
12.60 gdcmm::CompositeNetworkFunctions Class Reference	325
12.60.1 Detailed Description	326
12.60.2 Member Typedef Documentation	326
12.60.2.1 KeyValuePairArrayType	326
12.60.2.2 KeyValueType	327
12.60.3 Member Function Documentation	327
12.60.3.1 CEcho()	327
12.60.3.2 CFind()	327
12.60.3.3 CMove()	328
12.60.3.4 ConstructQuery() [1/2]	329
12.60.3.5 ConstructQuery() [2/2]	329
12.60.3.6 CStore()	329
12.61 gdcmm::ConstCharWrapper Class Reference	330
12.61.1 Detailed Description	330
12.61.2 Constructor & Destructor Documentation	330
12.61.2.1 ConstCharWrapper()	330
12.61.3 Member Function Documentation	331
12.61.3.1 operator const char *()	331
12.62 gdcmm::CP246ExplicitDataElement Class Reference	331
12.62.1 Detailed Description	334
12.62.2 Member Function Documentation	334
12.62.2.1 GetLength()	334
12.62.2.2 Read()	334
12.62.2.3 ReadPreValue()	334
12.62.2.4 ReadValue()	334
12.62.2.5 ReadWithLength()	335
12.63 gdcmm::CryptoFactory Class Reference	335
12.63.1 Detailed Description	336
12.63.2 Member Enumeration Documentation	336
12.63.2.1 CryptoLib	336

12.63.3 Constructor & Destructor Documentation	336
12.63.3.1 CryptoFactory() [1/2]	336
12.63.3.2 CryptoFactory() [2/2]	337
12.63.3.3 ~CryptoFactory()	337
12.63.4 Member Function Documentation	337
12.63.4.1 CreateCMSProvider()	337
12.63.4.2 GetFactoryInstance()	337
12.64 gdcmm::CryptographicMessageSyntax Class Reference	337
12.64.1 Member Enumeration Documentation	338
12.64.1.1 CipherTypes	338
12.64.2 Constructor & Destructor Documentation	338
12.64.2.1 CryptographicMessageSyntax() [1/2]	338
12.64.2.2 ~CryptographicMessageSyntax()	339
12.64.2.3 CryptographicMessageSyntax() [2/2]	339
12.64.3 Member Function Documentation	339
12.64.3.1 Decrypt()	339
12.64.3.2 Encrypt()	339
12.64.3.3 GetCipherType()	339
12.64.3.4 operator=()	340
12.64.3.5 ParseCertificateFile()	340
12.64.3.6 ParseKeyFile()	340
12.64.3.7 SetCipherType()	340
12.64.3.8 SetPassword()	340
12.65 gdcmm::CSAElement Class Reference	341
12.65.1 Detailed Description	342
12.65.2 Member Typedef Documentation	342
12.65.2.1 DataPtr	342
12.65.3 Constructor & Destructor Documentation	343
12.65.3.1 CSAElement() [1/2]	343
12.65.3.2 CSAElement() [2/2]	343
12.65.4 Member Function Documentation	343
12.65.4.1 GetByteValue()	343
12.65.4.2 GetKey()	343
12.65.4.3 GetName()	344
12.65.4.4 GetNoOfItems()	344
12.65.4.5 GetSyngoDT()	344
12.65.4.6 GetValue() [1/2]	344
12.65.4.7 GetValue() [2/2]	344
12.65.4.8 GetVM()	345
12.65.4.9 GetVR()	345

12.65.4.10	IsEmpty()	345
12.65.4.11	operator<()	345
12.65.4.12	operator=()	345
12.65.4.13	operator==()	346
12.65.4.14	SetByteValue()	346
12.65.4.15	SetKey()	346
12.65.4.16	SetName()	346
12.65.4.17	SetNoOfItems()	346
12.65.4.18	SetSyngoDT()	346
12.65.4.19	SetValue()	347
12.65.4.20	SetVM()	347
12.65.4.21	SetVR()	347
12.65.5	Friends And Related Symbol Documentation	347
12.65.5.1	operator<<	347
12.65.6	Member Data Documentation	347
12.65.6.1	DataField	347
12.65.6.2	KeyField	348
12.65.6.3	NameField	348
12.65.6.4	NoOfItemsField	348
12.65.6.5	SyngoDTField	348
12.65.6.6	ValueMultiplicityField	348
12.65.6.7	VRField	348
12.66	gdcm::CSAHeader Class Reference	348
12.66.1	Detailed Description	350
12.66.2	Member Enumeration Documentation	350
12.66.2.1	CSAHeaderType	350
12.66.3	Constructor & Destructor Documentation	351
12.66.3.1	CSAHeader()	351
12.66.3.2	~CSAHeader()	351
12.66.4	Member Function Documentation	351
12.66.4.1	FindCSAElementByName()	351
12.66.4.2	GetCSADDataInfo()	351
12.66.4.3	GetCSAEEnd()	352
12.66.4.4	GetCSAElementByName()	352
12.66.4.5	GetCSAImageHeaderInfoTag()	352
12.66.4.6	GetCSASeriesHeaderInfoTag()	352
12.66.4.7	GetDataSet()	352
12.66.4.8	GetFormat()	353
12.66.4.9	GetInterfile()	353
12.66.4.10	GetMrProtocol()	353

12.66.4.11 LoadFromDataElement()	353
12.66.4.12 Print()	353
12.66.5 Friends And Related Symbol Documentation	354
12.66.5.1 operator<<	354
12.67 gdcm::CSAHeaderDict Class Reference	354
12.67.1 Detailed Description	355
12.67.2 Member Typedef Documentation	355
12.67.2.1 ConstIterator	355
12.67.2.2 Iterator	355
12.67.2.3 MapCSAHeaderDictEntry	355
12.67.3 Constructor & Destructor Documentation	355
12.67.3.1 CSAHeaderDict() [1/2]	355
12.67.3.2 CSAHeaderDict() [2/2]	355
12.67.4 Member Function Documentation	356
12.67.4.1 AddCSAHeaderDictEntry()	356
12.67.4.2 Begin()	356
12.67.4.3 End()	356
12.67.4.4 GetCSAHeaderDictEntry()	356
12.67.4.5 IsEmpty()	356
12.67.4.6 LoadDefault()	356
12.67.4.7 operator=()	356
12.67.5 Friends And Related Symbol Documentation	357
12.67.5.1 Dicts	357
12.67.5.2 operator<<	357
12.68 gdcm::CSAHeaderDictEntry Class Reference	357
12.68.1 Detailed Description	358
12.68.2 Constructor & Destructor Documentation	358
12.68.2.1 CSAHeaderDictEntry()	358
12.68.3 Member Function Documentation	358
12.68.3.1 GetDescription()	358
12.68.3.2 GetName()	359
12.68.3.3 GetVM()	359
12.68.3.4 GetVR()	359
12.68.3.5 operator<()	359
12.68.3.6 SetDescription()	359
12.68.3.7 SetName()	359
12.68.3.8 SetVM()	359
12.68.3.9 SetVR()	360
12.68.4 Friends And Related Symbol Documentation	360
12.68.4.1 operator<<	360

12.69	gdcm::CSAHeaderDictException Class Reference	360
12.70	gdcm::network::CStoreRQ Class Reference	361
12.70.1	Detailed Description	362
12.70.2	Member Function Documentation	362
12.70.2.1	ConstructPDV()	362
12.71	gdcm::network::CStoreRSP Class Reference	363
12.71.1	Detailed Description	364
12.71.2	Member Function Documentation	364
12.71.2.1	ConstructPDV()	364
12.72	gdcm::Curve Class Reference	364
12.72.1	Detailed Description	366
12.72.2	Constructor & Destructor Documentation	366
12.72.2.1	Curve() [1/2]	366
12.72.2.2	~Curve()	366
12.72.2.3	Curve() [2/2]	366
12.72.3	Member Function Documentation	366
12.72.3.1	Decode()	366
12.72.3.2	GetAsPoints()	367
12.72.3.3	GetCurveDataDescriptor()	367
12.72.3.4	GetDataValueRepresentation()	367
12.72.3.5	GetDimensions()	367
12.72.3.6	GetGroup()	367
12.72.3.7	GetNumberOfCurves()	367
12.72.3.8	GetNumberOfPoints()	367
12.72.3.9	GetTypeOfData()	367
12.72.3.10	GetTypeOfDataDescription()	367
12.72.3.11	IsEmpty()	368
12.72.3.12	Print()	368
12.72.3.13	SetCoordinateStartValue()	368
12.72.3.14	SetCoordinateStepValue()	368
12.72.3.15	SetCurve()	368
12.72.3.16	SetCurveDataDescriptor()	368
12.72.3.17	SetCurveDescription()	368
12.72.3.18	SetDataValueRepresentation()	368
12.72.3.19	SetDimensions()	369
12.72.3.20	SetGroup()	369
12.72.3.21	SetNumberOfPoints()	369
12.72.3.22	SetTypeOfData()	369
12.72.3.23	Update()	369
12.73	gdcm::DataElement Class Reference	369

12.73.1 Detailed Description	372
12.73.2 Member Typedef Documentation	373
12.73.2.1 ValuePtr	373
12.73.3 Constructor & Destructor Documentation	373
12.73.3.1 DataElement() [1/2]	373
12.73.3.2 DataElement() [2/2]	373
12.73.4 Member Function Documentation	373
12.73.4.1 Clear()	373
12.73.4.2 Empty()	373
12.73.4.3 GetByteValue()	374
12.73.4.4 GetLength()	374
12.73.4.5 GetSequenceOffragments() [1/2]	374
12.73.4.6 GetSequenceOffragments() [2/2]	374
12.73.4.7 GetTag() [1/2]	375
12.73.4.8 GetTag() [2/2]	375
12.73.4.9 GetValue() [1/2]	375
12.73.4.10 GetValue() [2/2]	375
12.73.4.11 GetValueAsSQ()	376
12.73.4.12 GetVL() [1/2]	376
12.73.4.13 GetVL() [2/2]	376
12.73.4.14 GetVR()	377
12.73.4.15 IsEmpty()	377
12.73.4.16 IsUndefinedLength()	377
12.73.4.17 operator<()	378
12.73.4.18 operator=()	378
12.73.4.19 operator==()	378
12.73.4.20 Read()	378
12.73.4.21 ReadOrSkip()	378
12.73.4.22 ReadPreValue()	379
12.73.4.23 ReadValue()	379
12.73.4.24 ReadValueWithLength()	379
12.73.4.25 ReadWithLength()	379
12.73.4.26 SetByteValue()	380
12.73.4.27 SetTag()	380
12.73.4.28 SetValue()	381
12.73.4.29 SetValueFieldLength()	381
12.73.4.30 SetVL()	381
12.73.4.31 SetVLToUndefined()	381
12.73.4.32 SetVR()	382
12.73.4.33 Write()	382

12.73.5 Friends And Related Symbol Documentation	382
12.73.5.1 operator<<	382
12.73.6 Member Data Documentation	383
12.73.6.1 TagField	383
12.73.6.2 ValueField	383
12.73.6.3 ValueLengthField	383
12.73.6.4 VRField	383
12.74 gdcmm::DataElementException Class Reference	384
12.75 gdcmm::DataEvent Class Reference	384
12.75.1 Detailed Description	386
12.75.2 Member Typedef Documentation	386
12.75.2.1 Self	386
12.75.2.2 Superclass	386
12.75.3 Constructor & Destructor Documentation	386
12.75.3.1 DataEvent() [1/2]	386
12.75.3.2 ~DataEvent()	387
12.75.3.3 DataEvent() [2/2]	387
12.75.4 Member Function Documentation	387
12.75.4.1 CheckEvent()	387
12.75.4.2 GetData()	387
12.75.4.3 GetDataLength()	387
12.75.4.4 GetEventName()	387
12.75.4.5 MakeObject()	387
12.75.4.6 operator=()	388
12.75.4.7 SetData()	388
12.76 gdcmm::DataSet Class Reference	388
12.76.1 Detailed Description	390
12.76.2 Member Typedef Documentation	391
12.76.2.1 ConstIterator	391
12.76.2.2 DataElementSet	391
12.76.2.3 Iterator	391
12.76.2.4 SizeType	391
12.76.3 Member Function Documentation	391
12.76.3.1 Begin() [1/2]	391
12.76.3.2 Begin() [2/2]	392
12.76.3.3 Clear()	392
12.76.3.4 ComputeDataElement()	392
12.76.3.5 ComputeGroupLength()	392
12.76.3.6 End() [1/2]	392
12.76.3.7 End() [2/2]	392

12.76.3.8 FindDataElement() [1/2]	393
12.76.3.9 FindDataElement() [2/2]	393
12.76.3.10 FindNextDataElement()	393
12.76.3.11 GetDataElement() [1/2]	393
12.76.3.12 GetDataElement() [2/2]	394
12.76.3.13 GetDEEnd()	394
12.76.3.14 GetDES() [1/2]	394
12.76.3.15 GetDES() [2/2]	394
12.76.3.16 GetLength()	395
12.76.3.17 GetMediaStorage()	395
12.76.3.18 GetPrivateCreator()	395
12.76.3.19 GetPrivateTag()	395
12.76.3.20 Insert()	395
12.76.3.21 InsertDataElement()	396
12.76.3.22 IsEmpty()	396
12.76.3.23 operator>()	396
12.76.3.24 operator=()	396
12.76.3.25 operator[]()	396
12.76.3.26 Print()	396
12.76.3.27 Read()	397
12.76.3.28 ReadNested()	397
12.76.3.29 ReadSelectedPrivateTags()	397
12.76.3.30 ReadSelectedPrivateTagsWithLength()	397
12.76.3.31 ReadSelectedTags()	397
12.76.3.32 ReadSelectedTagsWithLength()	398
12.76.3.33 ReadUpToTag()	398
12.76.3.34 ReadUpToTagWithLength()	398
12.76.3.35 ReadWithLength()	398
12.76.3.36 Remove()	398
12.76.3.37 Replace()	399
12.76.3.38 ReplaceEmpty()	399
12.76.3.39 Size()	399
12.76.3.40 Write()	399
12.76.4 Friends And Related Symbol Documentation	400
12.76.4.1 CSAHeader	400
12.76.4.2 operator<<	400
12.77 gdcm::DataSetEvent Class Reference	400
12.77.1 Detailed Description	402
12.77.2 Member Typedef Documentation	402
12.77.2.1 Self	402

12.77.2.2 Superclass	402
12.77.3 Constructor & Destructor Documentation	402
12.77.3.1 DataSetEvent() [1/2]	402
12.77.3.2 ~DataSetEvent()	402
12.77.3.3 DataSetEvent() [2/2]	402
12.77.4 Member Function Documentation	403
12.77.4.1 CheckEvent()	403
12.77.4.2 GetDataSet()	403
12.77.4.3 GetEventName()	403
12.77.4.4 MakeObject()	403
12.77.4.5 operator=()	403
12.77.5 Member Data Documentation	403
12.77.5.1 m_DataSet	403
12.78 gdcm::DataSetHelper Class Reference	404
12.78.1 Detailed Description	404
12.78.2 Member Function Documentation	404
12.78.2.1 ComputeVR()	404
12.79 gdcm::Decoder Class Reference	405
12.79.1 Detailed Description	405
12.79.2 Constructor & Destructor Documentation	405
12.79.2.1 ~Decoder()	405
12.79.3 Member Function Documentation	406
12.79.3.1 CanDecode()	406
12.79.3.2 Decode()	406
12.79.3.3 DecodeByStreams()	406
12.80 gdcm::DefinedTerms Class Reference	406
12.80.1 Detailed Description	407
12.80.2 Constructor & Destructor Documentation	407
12.80.2.1 DefinedTerms()	407
12.81 gdcm::Defs Class Reference	407
12.81.1 Detailed Description	408
12.81.2 Constructor & Destructor Documentation	408
12.81.2.1 Defs() [1/2]	408
12.81.2.2 ~Defs()	408
12.81.2.3 Defs() [2/2]	408
12.81.3 Member Function Documentation	409
12.81.3.1 GetIODFromFile()	409
12.81.3.2 GetIODNameFromMediaStorage()	409
12.81.3.3 GetIODs() [1/2]	409
12.81.3.4 GetIODs() [2/2]	409

12.81.3.5	GetMacros() [1/2]	409
12.81.3.6	GetMacros() [2/2]	409
12.81.3.7	GetModules() [1/2]	410
12.81.3.8	GetModules() [2/2]	410
12.81.3.9	GetTypeFromTag()	410
12.81.3.10	IsEmpty()	410
12.81.3.11	LoadDefaults()	410
12.81.3.12	LoadFromFile()	410
12.81.3.13	operator=()	410
12.81.3.14	Verify() [1/2]	411
12.81.3.15	Verify() [2/2]	411
12.81.4	Friends And Related Symbol Documentation	411
12.81.4.1	Global	411
12.82	gdcm::DeltaEncodingCodec Class Reference	411
12.82.1	Detailed Description	414
12.82.2	Constructor & Destructor Documentation	414
12.82.2.1	DeltaEncodingCodec()	414
12.82.2.2	~DeltaEncodingCodec()	414
12.82.3	Member Function Documentation	414
12.82.3.1	CanDecode()	414
12.82.3.2	Decode() [1/2]	414
12.82.3.3	Decode() [2/2]	415
12.83	gdcm::DICOMDIR Class Reference	415
12.83.1	Detailed Description	415
12.83.2	Constructor & Destructor Documentation	415
12.83.2.1	DICOMDIR() [1/2]	415
12.83.2.2	DICOMDIR() [2/2]	415
12.84	gdcm::DICOMDIRGenerator Class Reference	416
12.84.1	Detailed Description	416
12.84.2	Member Typedef Documentation	417
12.84.2.1	FileNamesType	417
12.84.2.2	FilenameType	417
12.84.3	Constructor & Destructor Documentation	417
12.84.3.1	DICOMDIRGenerator()	417
12.84.3.2	~DICOMDIRGenerator()	417
12.84.4	Member Function Documentation	417
12.84.4.1	AddImageDirectoryRecord()	417
12.84.4.2	AddPatientDirectoryRecord()	418
12.84.4.3	AddSeriesDirectoryRecord()	418
12.84.4.4	AddStudyDirectoryRecord()	418

12.84.4.5	Generate()	418
12.84.4.6	GetFile()	418
12.84.4.7	GetScanner()	418
12.84.4.8	SetDescriptor()	418
12.84.4.9	SetFile()	419
12.84.4.10	SetFilenames()	419
12.84.4.11	SetRootDirectory()	419
12.85	gdcm::Dict Class Reference	419
12.85.1	Detailed Description	420
12.85.2	Member Typedef Documentation	420
12.85.2.1	ConstIterator	420
12.85.2.2	Iterator	421
12.85.2.3	MapDictEntry	421
12.85.3	Constructor & Destructor Documentation	421
12.85.3.1	Dict() [1/2]	421
12.85.3.2	Dict() [2/2]	421
12.85.4	Member Function Documentation	421
12.85.4.1	AddDictEntry()	421
12.85.4.2	Begin()	421
12.85.4.3	End()	422
12.85.4.4	GetDictEntry()	422
12.85.4.5	GetDictEntryByKeyword()	422
12.85.4.6	GetDictEntryByName()	422
12.85.4.7	GetKeywordFromTag()	423
12.85.4.8	IsEmpty()	423
12.85.4.9	LoadDefault()	423
12.85.4.10	operator=()	423
12.85.5	Friends And Related Symbol Documentation	423
12.85.5.1	Dicts	423
12.85.5.2	operator<<	423
12.86	gdcm::DictConverter Class Reference	424
12.86.1	Detailed Description	425
12.86.2	Member Enumeration Documentation	425
12.86.2.1	OutputTypes	425
12.86.3	Constructor & Destructor Documentation	425
12.86.3.1	DictConverter()	425
12.86.3.2	~DictConverter()	425
12.86.4	Member Function Documentation	425
12.86.4.1	AddGroupLength()	425
12.86.4.2	Convert()	426

12.86.4.3	ConvertToCXX()	426
12.86.4.4	ConvertToXML()	426
12.86.4.5	GetDictName()	426
12.86.4.6	GetInputFilename()	426
12.86.4.7	GetOutputFilename()	426
12.86.4.8	GetOutputType()	426
12.86.4.9	Readuint16()	426
12.86.4.10	ReadVM()	427
12.86.4.11	ReadVR()	427
12.86.4.12	SetDictName()	427
12.86.4.13	SetInputFileName()	427
12.86.4.14	SetOutputFileName()	427
12.86.4.15	SetOutputType()	427
12.86.4.16	WriteFooter()	427
12.86.4.17	WriteHeader()	427
12.87	gdcm::DictEntry Class Reference	428
12.87.1	Detailed Description	428
12.87.2	Constructor & Destructor Documentation	429
12.87.2.1	DictEntry()	429
12.87.3	Member Function Documentation	429
12.87.3.1	GetKeyword()	429
12.87.3.2	GetName()	429
12.87.3.3	GetRetired()	429
12.87.3.4	GetVM()	429
12.87.3.5	GetVR()	430
12.87.3.6	IsUnique()	430
12.87.3.7	SetElementXX()	430
12.87.3.8	SetGroupXX()	430
12.87.3.9	SetKeyword()	430
12.87.3.10	SetName()	430
12.87.3.11	SetRetired()	431
12.87.3.12	SetVM()	431
12.87.3.13	SetVR()	431
12.87.4	Friends And Related Symbol Documentation	431
12.87.4.1	Dict	431
12.87.4.2	operator<<	431
12.88	gdcm::DictPrinter Class Reference	432
12.88.1	Detailed Description	434
12.88.2	Constructor & Destructor Documentation	434
12.88.2.1	DictPrinter()	434

12.88.2.2	~DictPrinter()	434
12.88.3	Member Function Documentation	434
12.88.3.1	Print()	434
12.88.3.2	PrintDataElement2()	434
12.88.3.3	PrintDataSet2()	434
12.89	gdcm::Dicts Class Reference	435
12.89.1	Detailed Description	436
12.89.2	Member Enumeration Documentation	436
12.89.2.1	ConstructorType	436
12.89.3	Constructor & Destructor Documentation	436
12.89.3.1	Dicts() [1/2]	436
12.89.3.2	~Dicts()	436
12.89.3.3	Dicts() [2/2]	436
12.89.4	Member Function Documentation	437
12.89.4.1	GetConstructorString()	437
12.89.4.2	GetCSAHeaderDict()	437
12.89.4.3	GetDictEntry() [1/2]	437
12.89.4.4	GetDictEntry() [2/2]	437
12.89.4.5	GetPrivateDict() [1/2]	437
12.89.4.6	GetPrivateDict() [2/2]	438
12.89.4.7	GetPublicDict()	438
12.89.4.8	IsEmpty()	438
12.89.4.9	LoadDefaults()	438
12.89.4.10	operator=()	438
12.89.5	Friends And Related Symbol Documentation	438
12.89.5.1	Global	438
12.89.5.2	operator<<	439
12.90	gdcm::network::DIMSE Class Reference	439
12.90.1	Detailed Description	439
12.90.2	Member Enumeration Documentation	440
12.90.2.1	CommandTypes	440
12.91	gdcm::DirectionCosines Class Reference	440
12.91.1	Detailed Description	441
12.91.2	Constructor & Destructor Documentation	441
12.91.2.1	DirectionCosines() [1/2]	441
12.91.2.2	DirectionCosines() [2/2]	442
12.91.2.3	~DirectionCosines()	442
12.91.3	Member Function Documentation	442
12.91.3.1	ComputeDistAlongNormal()	442
12.91.3.2	Cross()	442

12.91.3.3	CrossDot()	442
12.91.3.4	Dot() [1/2]	442
12.91.3.5	Dot() [2/2]	443
12.91.3.6	IsValid()	443
12.91.3.7	Norm()	443
12.91.3.8	Normalize() [1/2]	443
12.91.3.9	Normalize() [2/2]	443
12.91.3.10	operator const double *()	443
12.91.3.11	Print()	443
12.91.3.12	SetFromString()	444
12.92	gdcm::Directory Class Reference	444
12.92.1	Detailed Description	445
12.92.2	Member Typedef Documentation	445
12.92.2.1	FileNamesType	445
12.92.2.2	FilenameType	445
12.92.3	Constructor & Destructor Documentation	445
12.92.3.1	Directory()	445
12.92.3.2	~Directory()	446
12.92.4	Member Function Documentation	446
12.92.4.1	Explore()	446
12.92.4.2	GetDirectories()	446
12.92.4.3	GetFileNames()	446
12.92.4.4	GetToplevel()	446
12.92.4.5	Load()	447
12.92.4.6	Print()	447
12.92.5	Friends And Related Symbol Documentation	447
12.92.5.1	operator<<	447
12.93	gdcm::DirectoryHelper Class Reference	448
12.93.1	Detailed Description	448
12.93.2	Member Function Documentation	448
12.93.2.1	GetCTImageSeriesUIDs()	448
12.93.2.2	GetFileNamesFromSeriesUIDs()	449
12.93.2.3	GetFrameOfReference()	449
12.93.2.4	GetMRImageSeriesUIDs()	449
12.93.2.5	GetRTStructSeriesUIDs()	449
12.93.2.6	GetSeriesUIDsBySOPClassUID()	449
12.93.2.7	GetSOPClassUID()	449
12.93.2.8	GetStringValueFromTag()	450
12.93.2.9	LoadImageFromFiles()	450
12.93.2.10	RetrieveSOPInstanceUIDFromIndex()	450

12.93.2.11 RetrieveSOPInstanceUIDFromZPosition()	450
12.94 gdcm::DPath Class Reference	450
12.94.1 Detailed Description	451
12.94.2 Constructor & Destructor Documentation	451
12.94.2.1 DPath()	451
12.94.2.2 ~DPath()	451
12.94.3 Member Function Documentation	451
12.94.3.1 ConstructFromString()	451
12.94.3.2 IsValid()	451
12.94.3.3 Match()	452
12.94.3.4 operator<()	452
12.94.3.5 Print()	452
12.94.4 Friends And Related Symbol Documentation	452
12.94.4.1 operator<<	452
12.95 gdcm::DummyValueGenerator Class Reference	452
12.95.1 Detailed Description	453
12.95.2 Member Function Documentation	453
12.95.2.1 Generate()	453
12.96 gdcm::Dumper Class Reference	453
12.96.1 Detailed Description	455
12.96.2 Constructor & Destructor Documentation	455
12.96.2.1 Dumper()	455
12.96.2.2 ~Dumper()	455
12.97 gdcm::Element< TVR, TVM > Class Template Reference	456
12.97.1 Detailed Description	458
12.97.2 Member Typedef Documentation	458
12.97.2.1 Type	458
12.97.3 Member Function Documentation	458
12.97.3.1 GetAsDataElement()	458
12.97.3.2 GetLength()	459
12.97.3.3 GetValue() [1/2]	459
12.97.3.4 GetValue() [2/2]	459
12.97.3.5 GetValues()	459
12.97.3.6 GetVM()	459
12.97.3.7 GetVR()	460
12.97.3.8 operator[]()	460
12.97.3.9 Print()	460
12.97.3.10 Read()	460
12.97.3.11 Set()	460
12.97.3.12 SetFromDataElement()	461

12.97.3.13 SetNoSwap()	461
12.97.3.14 SetValue()	461
12.97.3.15 Write()	461
12.97.4 Member Data Documentation	462
12.97.4.1 Internal	462
12.98 gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	462
12.98.1 Member Typedef Documentation	465
12.98.1.1 Parent	465
12.98.1.2 Type	465
12.98.2 Member Function Documentation	465
12.98.2.1 GetAsDataElement()	465
12.98.2.2 GetLength()	465
12.98.2.3 GetValue()	465
12.98.2.4 GetValues()	465
12.98.2.5 GetVM()	466
12.98.2.6 GetVR()	466
12.98.2.7 operator[]()	466
12.98.2.8 Print()	466
12.98.2.9 Read()	466
12.98.2.10 Set()	466
12.98.2.11 SetFromDataElement()	466
12.98.2.12 SetLength()	466
12.98.2.13 SetNoSwap()	467
12.98.2.14 SetValue()	467
12.98.2.15 Write()	467
12.98.3 Member Data Documentation	467
12.98.3.1 Internal	467
12.99 gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	467
12.99.1 Member Typedef Documentation	469
12.99.1.1 Type	469
12.99.2 Constructor & Destructor Documentation	469
12.99.2.1 Element() [1/2]	469
12.99.2.2 ~Element()	469
12.99.2.3 Element() [2/2]	469
12.99.3 Member Function Documentation	470
12.99.3.1 GetAsDataElement()	470
12.99.3.2 GetLength()	470
12.99.3.3 GetValue() [1/2]	470
12.99.3.4 GetValue() [2/2]	470
12.99.3.5 GetValues()	470

12.99.3.6	GetVM()	470
12.99.3.7	GetVR()	471
12.99.3.8	operator=()	471
12.99.3.9	operator[]()	471
12.99.3.10	Print()	471
12.99.3.11	Read()	471
12.99.3.12	Set()	472
12.99.3.13	SetArray()	472
12.99.3.14	SetFromDataElement()	472
12.99.3.15	SetLength()	472
12.99.3.16	SetNoSwap()	473
12.99.3.17	SetValue()	473
12.99.3.18	Write()	473
12.99.3.19	WriteASCII()	473
12.100	gdcmm::Element< TVR, VM::VM2_2n > Class Template Reference	474
12.100.1	Member Typedef Documentation	477
12.100.1.1	Parent	477
12.100.1.2	Type	478
12.100.2	Member Function Documentation	478
12.100.2.1	GetAsDataElement()	478
12.100.2.2	GetLength()	478
12.100.2.3	GetValue()	478
12.100.2.4	GetValues()	478
12.100.2.5	GetVM()	478
12.100.2.6	GetVR()	478
12.100.2.7	operator[]()	478
12.100.2.8	Print()	478
12.100.2.9	Read()	479
12.100.2.10	Set()	479
12.100.2.11	SetFromDataElement()	479
12.100.2.12	SetLength()	479
12.100.2.13	SetNoSwap()	479
12.100.2.14	SetValue()	479
12.100.2.15	Write()	479
12.100.3	Member Data Documentation	480
12.100.3.1	Internal	480
12.101	gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	480
12.101.1	Member Typedef Documentation	483
12.101.1.1	Parent	483
12.101.1.2	Type	483

12.101.2 Member Function Documentation	483
12.101.2.1 GetAsDataElement()	483
12.101.2.2 GetLength()	483
12.101.2.3 GetValue()	483
12.101.2.4 GetValues()	483
12.101.2.5 GetVM()	484
12.101.2.6 GetVR()	484
12.101.2.7 operator[]()	484
12.101.2.8 Print()	484
12.101.2.9 Read()	484
12.101.2.10 Set()	484
12.101.2.11 SetFromDataElement()	484
12.101.2.12 SetLength()	484
12.101.2.13 SetNoSwap()	485
12.101.2.14 SetValue()	485
12.101.2.15 Write()	485
12.101.3 Member Data Documentation	485
12.101.3.1 Internal	485
12.102 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference	485
12.102.1 Member Typedef Documentation	489
12.102.1.1 Parent	489
12.102.1.2 Type	490
12.102.2 Member Function Documentation	490
12.102.2.1 GetAsDataElement()	490
12.102.2.2 GetLength()	490
12.102.2.3 GetValue()	490
12.102.2.4 GetValues()	490
12.102.2.5 GetVM()	490
12.102.2.6 GetVR()	490
12.102.2.7 operator[]()	490
12.102.2.8 Print()	490
12.102.2.9 Read()	491
12.102.2.10 Set()	491
12.102.2.11 SetFromDataElement()	491
12.102.2.12 SetLength()	491
12.102.2.13 SetNoSwap()	491
12.102.2.14 SetValue()	491
12.102.2.15 Write()	491
12.102.3 Member Data Documentation	492
12.102.3.1 Internal	492

12.103 gdcmm::Element< TVR, VM::VM3_4 > Class Template Reference	492
12.103.1 Member Typedef Documentation	495
12.103.1.1 Parent	495
12.103.1.2 Type	495
12.103.2 Member Function Documentation	495
12.103.2.1 GetAsDataElement()	495
12.103.2.2 GetLength()	495
12.103.2.3 GetValue()	495
12.103.2.4 GetValues()	495
12.103.2.5 GetVM()	496
12.103.2.6 GetVR()	496
12.103.2.7 operator[]()	496
12.103.2.8 Print()	496
12.103.2.9 Read()	496
12.103.2.10 Set()	496
12.103.2.11 SetFromDataElement()	496
12.103.2.12 SetLength()	496
12.103.2.13 SetNoSwap()	497
12.103.2.14 SetValue()	497
12.103.2.15 Write()	497
12.103.3 Member Data Documentation	497
12.103.3.1 Internal	497
12.104 gdcmm::Element< TVR, VM::VM3_n > Class Template Reference	497
12.104.1 Member Typedef Documentation	501
12.104.1.1 Parent	501
12.104.1.2 Type	501
12.104.2 Member Function Documentation	501
12.104.2.1 GetAsDataElement()	501
12.104.2.2 GetLength()	501
12.104.2.3 GetValue()	501
12.104.2.4 GetValues()	501
12.104.2.5 GetVM()	502
12.104.2.6 GetVR()	502
12.104.2.7 operator[]()	502
12.104.2.8 Print()	502
12.104.2.9 Read()	502
12.104.2.10 Set()	502
12.104.2.11 SetFromDataElement()	502
12.104.2.12 SetLength()	502
12.104.2.13 SetNoSwap()	503

12.104.2.14 SetValue()	503
12.104.2.15 Write()	503
12.104.3 Member Data Documentation	503
12.104.3.1 Internal	503
12.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference	503
12.105.1 Member Typedef Documentation	505
12.105.1.1 Type	505
12.105.2 Member Function Documentation	505
12.105.2.1 GetAsDataElement()	505
12.105.2.2 GetLength()	505
12.105.2.3 GetValue()	505
12.105.2.4 GetValues()	505
12.105.2.5 GetVM()	505
12.105.2.6 GetVR()	506
12.105.2.7 operator[]()	506
12.105.2.8 Print()	506
12.105.2.9 Read()	506
12.105.2.10 Set()	506
12.105.2.11 SetFromDataElement()	506
12.105.2.12 SetNoSwap()	506
12.105.2.13 SetValue()	506
12.105.2.14 Write()	507
12.105.3 Member Data Documentation	507
12.105.3.1 Internal	507
12.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference	507
12.106.1 Member Typedef Documentation	510
12.106.1.1 Type	510
12.106.2 Member Function Documentation	510
12.106.2.1 GetAsDataElement()	510
12.106.2.2 GetLength()	510
12.106.2.3 GetValue()	510
12.106.2.4 GetValues()	510
12.106.2.5 GetVM()	510
12.106.2.6 GetVR()	510
12.106.2.7 operator[]()	511
12.106.2.8 Print()	511
12.106.2.9 Read()	511
12.106.2.10 Set()	511
12.106.2.11 SetFromDataElement()	511
12.106.2.12 SetNoSwap()	511

12.106.2.13 SetValue()	511
12.106.2.14 Write()	511
12.106.3 Member Data Documentation	512
12.106.3.1 Internal	512
12.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference	512
12.107.1 Member Typedef Documentation	515
12.107.1.1 Type	515
12.107.2 Member Function Documentation	515
12.107.2.1 GetAsDataElement()	515
12.107.2.2 GetLength()	515
12.107.2.3 GetValue()	515
12.107.2.4 GetValues()	515
12.107.2.5 GetVM()	515
12.107.2.6 GetVR()	515
12.107.2.7 operator[]()	516
12.107.2.8 Print()	516
12.107.2.9 Read()	516
12.107.2.10 Set()	516
12.107.2.11 SetFromDataElement()	516
12.107.2.12 SetNoSwap()	516
12.107.2.13 SetValue()	516
12.107.2.14 Write()	516
12.107.3 Member Data Documentation	517
12.107.3.1 Internal	517
12.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference	517
12.108.1 Detailed Description	517
12.109 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference	518
12.110 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference	519
12.111 gdcm::EmptyMaskGenerator Class Reference	520
12.111.1 Detailed Description	520
12.111.2 Member Enumeration Documentation	521
12.111.2.1 SOPClassUIDMode	521
12.111.3 Constructor & Destructor Documentation	521
12.111.3.1 EmptyMaskGenerator()	521
12.111.3.2 ~EmptyMaskGenerator()	521
12.111.4 Member Function Documentation	521
12.111.4.1 Execute()	521
12.111.4.2 SetInputDirectory()	522
12.111.4.3 SetOutputDirectory()	522
12.111.4.4 SetSOPClassUIDMode()	522

12.112	gdcm::EncapsulatedDocument Class Reference	522
12.112.1	Detailed Description	523
12.112.2	Constructor & Destructor Documentation	523
12.112.2.1	EncapsulatedDocument()	523
12.113	gdcm::EncodingImplementation< T > Class Template Reference	523
12.113.1	Detailed Description	523
12.114	gdcm::EncodingImplementation< VR::VRASCII > Class Reference	524
12.114.1	Member Function Documentation	525
12.114.1.1	Read()	525
12.114.1.2	ReadComputeLength()	525
12.114.1.3	ReadNoSwap()	525
12.114.1.4	Write() [1/2]	526
12.114.1.5	Write() [2/2]	526
12.115	gdcm::EncodingImplementation< VR::VRBINARY > Class Reference	526
12.115.1	Member Function Documentation	527
12.115.1.1	Read()	527
12.115.1.2	ReadComputeLength()	527
12.115.1.3	ReadNoSwap()	528
12.115.1.4	Write()	528
12.116	gdcm::EndEvent Class Reference	528
12.117	gdcm::EnumeratedValues Class Reference	529
12.117.1	Detailed Description	530
12.117.2	Constructor & Destructor Documentation	530
12.117.2.1	EnumeratedValues()	530
12.118	gdcm::EquipmentManufacturer Class Reference	530
12.118.1	Detailed Description	531
12.118.2	Member Enumeration Documentation	531
12.118.2.1	Type	531
12.118.3	Member Function Documentation	531
12.118.3.1	Compute()	531
12.118.3.2	TypeToString()	532
12.119	gdcm::Event Class Reference	532
12.119.1	Detailed Description	533
12.119.2	Constructor & Destructor Documentation	533
12.119.2.1	Event() [1/2]	533
12.119.2.2	~Event()	533
12.119.2.3	Event() [2/2]	533
12.119.3	Member Function Documentation	534
12.119.3.1	CheckEvent()	534
12.119.3.2	GetEventName()	534

12.119.3.3	MakeObject()	534
12.119.3.4	operator=()	534
12.119.3.5	Print()	535
12.120	gdcmm::Exception Class Reference	535
12.120.1	Detailed Description	536
12.120.2	Constructor & Destructor Documentation	537
12.120.2.1	Exception()	537
12.120.2.2	~Exception()	537
12.120.3	Member Function Documentation	537
12.120.3.1	GetDescription()	537
12.120.3.2	what()	537
12.121	gdcmm::ExitEvent Class Reference	538
12.122	gdcmm::ExplicitDataElement Class Reference	539
12.122.1	Detailed Description	542
12.122.2	Member Function Documentation	542
12.122.2.1	GetLength()	542
12.122.2.2	Read()	542
12.122.2.3	ReadPreValue()	542
12.122.2.4	ReadValue()	543
12.122.2.5	ReadWithLength()	543
12.122.2.6	Write()	543
12.123	gdcmm::ExplicitImplicitDataElement Class Reference	543
12.123.1	Detailed Description	546
12.123.2	Member Function Documentation	546
12.123.2.1	GetLength()	546
12.123.2.2	Read()	546
12.123.2.3	ReadPreValue()	546
12.123.2.4	ReadValue()	546
12.123.2.5	ReadWithLength()	547
12.124	gdcmm::Fiducials Class Reference	547
12.124.1	Detailed Description	547
12.124.2	Constructor & Destructor Documentation	547
12.124.2.1	Fiducials()	547
12.125	gdcmm::File Class Reference	548
12.125.1	Detailed Description	549
12.125.2	Constructor & Destructor Documentation	550
12.125.2.1	File()	550
12.125.2.2	~File()	550
12.125.3	Member Function Documentation	550
12.125.3.1	GetDataSet() [1/2]	550

12.125.3.2	GetDataSet() [2/2]	551
12.125.3.3	GetHeader() [1/2]	551
12.125.3.4	GetHeader() [2/2]	551
12.125.3.5	Read()	551
12.125.3.6	SetDataSet()	552
12.125.3.7	SetHeader()	552
12.125.3.8	Write()	552
12.125.4	Friends And Related Symbol Documentation	552
12.125.4.1	operator<<	552
12.126	gdcmm::FileAnonymizer Class Reference	553
12.126.1	Detailed Description	555
12.126.2	Constructor & Destructor Documentation	555
12.126.2.1	FileAnonymizer()	555
12.126.2.2	~FileAnonymizer()	555
12.126.3	Member Function Documentation	555
12.126.3.1	Empty()	555
12.126.3.2	Remove()	556
12.126.3.3	Replace() [1/2]	556
12.126.3.4	Replace() [2/2]	556
12.126.3.5	SetInputFileName()	556
12.126.3.6	SetOutputFileName()	557
12.126.3.7	Write()	557
12.127	gdcmm::FileChangeTransferSyntax Class Reference	557
12.127.1	Detailed Description	559
12.127.2	Constructor & Destructor Documentation	559
12.127.2.1	FileChangeTransferSyntax()	559
12.127.2.2	~FileChangeTransferSyntax()	560
12.127.3	Member Function Documentation	560
12.127.3.1	Change()	560
12.127.3.2	GetCodec()	560
12.127.3.3	New()	560
12.127.3.4	SetInputFileName()	560
12.127.3.5	SetOutputFileName()	561
12.127.3.6	SetTransferSyntax()	561
12.128	gdcmm::FileDecompressLookupTable Class Reference	561
12.128.1	Detailed Description	563
12.128.2	Constructor & Destructor Documentation	563
12.128.2.1	FileDecompressLookupTable()	563
12.128.2.2	~FileDecompressLookupTable()	563
12.128.3	Member Function Documentation	563

12.128.3.1	Change()	563
12.128.3.2	GetFile()	564
12.128.3.3	GetPixmap() [1/2]	564
12.128.3.4	GetPixmap() [2/2]	564
12.128.3.5	SetFile()	564
12.128.3.6	SetPixmap()	564
12.129	gdcm::FileDerivation Class Reference	564
12.129.1	Detailed Description	565
12.129.2	Constructor & Destructor Documentation	565
12.129.2.1	FileDerivation()	565
12.129.2.2	~FileDerivation()	566
12.129.3	Member Function Documentation	566
12.129.3.1	AddDerivationDescription()	566
12.129.3.2	AddPurposeOfReferenceCodeSequence()	566
12.129.3.3	AddReference()	566
12.129.3.4	AddSourceImageSequence()	566
12.129.3.5	Derive()	566
12.129.3.6	GetFile() [1/2]	567
12.129.3.7	GetFile() [2/2]	567
12.129.3.8	SetAppendDerivationHistory()	567
12.129.3.9	SetDerivationCodeSequenceCodeValue()	567
12.129.3.10	SetDerivationDescription()	567
12.129.3.11	SetFile()	568
12.129.3.12	SetPurposeOfReferenceCodeSequenceCodeValue()	568
12.130	gdcm::FileExplicitFilter Class Reference	568
12.130.1	Detailed Description	569
12.130.2	Constructor & Destructor Documentation	569
12.130.2.1	FileExplicitFilter()	569
12.130.2.2	~FileExplicitFilter()	569
12.130.3	Member Function Documentation	569
12.130.3.1	Change()	569
12.130.3.2	ChangeFMI()	570
12.130.3.3	GetFile()	570
12.130.3.4	ProcessDataSet()	570
12.130.3.5	SetChangePrivateTags()	570
12.130.3.6	SetFile()	570
12.130.3.7	SetRecomputeItemLength()	571
12.130.3.8	SetRecomputeSequenceLength()	571
12.130.3.9	SetUseVRUN()	571
12.131	gdcm::FileMetaInformation Class Reference	571

12.131.1 Detailed Description	575
12.131.2 Constructor & Destructor Documentation	575
12.131.2.1 FileMetaInformation() [1/2]	575
12.131.2.2 ~FileMetaInformation()	576
12.131.2.3 FileMetaInformation() [2/2]	576
12.131.3 Member Function Documentation	576
12.131.3.1 AppendImplementationClassUID()	576
12.131.3.2 ComputeDataSetMediaStorageSOPClass()	576
12.131.3.3 ComputeDataSetTransferSyntax()	576
12.131.3.4 Default()	576
12.131.3.5 FillFromDataSet()	576
12.131.3.6 GetDataSetTransferSyntax()	577
12.131.3.7 GetFileMetaInformationVersion()	577
12.131.3.8 GetFullLength()	577
12.131.3.9 GetGDCMImplementationClassUID()	577
12.131.3.10 GetGDCMImplementationVersionName()	577
12.131.3.11 GetGDCMSourceApplicationEntityTitle()	577
12.131.3.12 GetImplementationClassUID()	577
12.131.3.13 GetImplementationVersionName()	577
12.131.3.14 GetMediaStorage()	578
12.131.3.15 GetMediaStorageAsString()	578
12.131.3.16 GetMetaInformationTS()	578
12.131.3.17 GetPreamble() [1/2]	578
12.131.3.18 GetPreamble() [2/2]	578
12.131.3.19 GetSourceApplicationEntityTitle()	578
12.131.3.20 Insert()	578
12.131.3.21 IsValid()	578
12.131.3.22 operator=()	579
12.131.3.23 Read()	579
12.131.3.24 ReadCompat()	579
12.131.3.25 ReadCompatInternal()	579
12.131.3.26 Replace()	579
12.131.3.27 SetDataSetTransferSyntax()	579
12.131.3.28 SetImplementationClassUID()	580
12.131.3.29 SetImplementationVersionName()	580
12.131.3.30 SetPreamble()	580
12.131.3.31 SetSourceApplicationEntityTitle()	580
12.131.3.32 Write()	580
12.131.4 Friends And Related Symbol Documentation	580
12.131.4.1 operator<<	580

12.131.5 Member Data Documentation	581
12.131.5.1 DataSetMS	581
12.131.5.2 DataSetTS	581
12.131.5.3 MetaInformationTS	581
12.132 gdcm::Filename Class Reference	581
12.132.1 Detailed Description	582
12.132.2 Constructor & Destructor Documentation	582
12.132.2.1 Filename()	582
12.132.3 Member Function Documentation	582
12.132.3.1 EndWith()	582
12.132.3.2 GetExtension()	582
12.132.3.3 GetFileName()	582
12.132.3.4 GetName()	583
12.132.3.5 GetPath()	583
12.132.3.6 IsEmpty()	583
12.132.3.7 IsIdentical()	583
12.132.3.8 Join()	583
12.132.3.9 operator const char *()	583
12.132.3.10 ToUnixSlashes()	584
12.132.3.11 ToWindowsSlashes()	584
12.133 gdcm::FileNameEvent Class Reference	584
12.133.1 Detailed Description	586
12.133.2 Member Typedef Documentation	586
12.133.2.1 Self	586
12.133.2.2 Superclass	586
12.133.3 Constructor & Destructor Documentation	586
12.133.3.1 FileNameEvent() [1/2]	586
12.133.3.2 ~FileNameEvent()	586
12.133.3.3 FileNameEvent() [2/2]	586
12.133.4 Member Function Documentation	587
12.133.4.1 CheckEvent()	587
12.133.4.2 GetEventName()	587
12.133.4.3 GetFileName()	587
12.133.4.4 MakeObject()	587
12.133.4.5 operator=()	587
12.133.4.6 SetFileName()	587
12.134 gdcm::FilenameGenerator Class Reference	588
12.134.1 Detailed Description	588
12.134.2 Member Typedef Documentation	589
12.134.2.1 FilenamesType	589

12.134.2.2	FilenameType	589
12.134.2.3	SizeType	589
12.134.3	Constructor & Destructor Documentation	589
12.134.3.1	FilenameGenerator()	589
12.134.3.2	~FilenameGenerator()	589
12.134.4	Member Function Documentation	589
12.134.4.1	Generate()	589
12.134.4.2	GetFilename()	590
12.134.4.3	GetFileNames()	590
12.134.4.4	GetNumberOfFileNames()	590
12.134.4.5	GetPattern()	590
12.134.4.6	GetPrefix()	590
12.134.4.7	SetNumberOfFileNames()	590
12.134.4.8	SetPattern()	591
12.134.4.9	SetPrefix()	591
12.135	gdcm::FileSet Class Reference	591
12.135.1	Detailed Description	592
12.135.2	Member Typedef Documentation	592
12.135.2.1	FileType	592
12.135.2.2	FileType	592
12.135.3	Constructor & Destructor Documentation	592
12.135.3.1	FileSet()	592
12.135.4	Member Function Documentation	592
12.135.4.1	AddFile() [1/2]	592
12.135.4.2	AddFile() [2/2]	592
12.135.4.3	GetFiles()	593
12.135.4.4	SetFiles()	593
12.135.5	Friends And Related Symbol Documentation	593
12.135.5.1	operator<<	593
12.136	gdcm::FileStreamer Class Reference	593
12.136.1	Detailed Description	595
12.136.2	Constructor & Destructor Documentation	596
12.136.2.1	FileStreamer()	596
12.136.2.2	~FileStreamer()	596
12.136.3	Member Function Documentation	596
12.136.3.1	AppendToDataElement()	596
12.136.3.2	AppendToGroupDataElement()	596
12.136.3.3	CheckDataElement()	596
12.136.3.4	CheckTemplateFileName()	597
12.136.3.5	New()	597

12.136.3.6 ReserveDataElement()	597
12.136.3.7 ReserveGroupDataElement()	597
12.136.3.8 SetOutputFileName()	597
12.136.3.9 SetTemplateFileName()	598
12.136.3.10 StartDataElement()	598
12.136.3.11 StartGroupDataElement()	598
12.136.3.12 StopDataElement()	598
12.136.3.13 StopGroupDataElement()	599
12.137 gdcm::FileWithName Class Reference	599
12.137.1 Detailed Description	601
12.137.2 Constructor & Destructor Documentation	601
12.137.2.1 FileWithName()	601
12.137.3 Member Data Documentation	601
12.137.3.1 filename	601
12.138 gdcm::FindPatientRootQuery Class Reference	602
12.138.1 Detailed Description	604
12.138.2 Constructor & Destructor Documentation	604
12.138.2.1 FindPatientRootQuery()	604
12.138.3 Member Function Documentation	604
12.138.3.1 GetAbstractSyntaxUID()	604
12.138.3.2 GetTagListByLevel()	605
12.138.3.3 InitializeDataSet()	605
12.138.3.4 ValidateQuery()	605
12.138.4 Friends And Related Symbol Documentation	605
12.138.4.1 QueryFactory	605
12.139 gdcm::FindStudyRootQuery Class Reference	606
12.139.1 Detailed Description	608
12.139.2 Constructor & Destructor Documentation	608
12.139.2.1 FindStudyRootQuery()	608
12.139.3 Member Function Documentation	608
12.139.3.1 GetAbstractSyntaxUID()	608
12.139.3.2 GetTagListByLevel()	609
12.139.3.3 InitializeDataSet()	609
12.139.3.4 ValidateQuery()	609
12.139.4 Friends And Related Symbol Documentation	609
12.139.4.1 QueryFactory	609
12.140 gdcm::Fragment Class Reference	610
12.140.1 Detailed Description	612
12.140.2 Constructor & Destructor Documentation	613
12.140.2.1 Fragment()	613

12.140.3 Member Function Documentation	613
12.140.3.1 ComputeLength()	613
12.140.3.2 GetLength()	613
12.140.3.3 Read()	613
12.140.3.4 ReadBacktrack()	613
12.140.3.5 ReadPreValue()	614
12.140.3.6 ReadValue()	614
12.140.3.7 Write()	614
12.140.4 Friends And Related Symbol Documentation	614
12.140.4.1 operator<<	614
12.141 gdcmm::Global Class Reference	615
12.141.1 Detailed Description	615
12.141.2 Constructor & Destructor Documentation	616
12.141.2.1 Global() [1/2]	616
12.141.2.2 ~Global()	616
12.141.2.3 Global() [2/2]	616
12.141.3 Member Function Documentation	616
12.141.3.1 Append()	616
12.141.3.2 GetDefs()	616
12.141.3.3 GetDicts() [1/2]	617
12.141.3.4 GetDicts() [2/2]	617
12.141.3.5 GetInstance()	617
12.141.3.6 LoadResourcesFiles()	617
12.141.3.7 Locate()	618
12.141.3.8 operator=()	618
12.141.3.9 Prepend()	618
12.141.4 Friends And Related Symbol Documentation	618
12.141.4.1 operator<<	618
12.142 gdcmm::GroupDict Class Reference	618
12.142.1 Detailed Description	619
12.142.2 Member Typedef Documentation	619
12.142.2.1 GroupStringVector	619
12.142.3 Constructor & Destructor Documentation	619
12.142.3.1 GroupDict()	619
12.142.3.2 ~GroupDict()	620
12.142.4 Member Function Documentation	620
12.142.4.1 Add()	620
12.142.4.2 GetAbbreviation()	620
12.142.4.3 GetName()	620
12.142.4.4 Insert()	620

12.142.4.5	Size()	620
12.142.5	Friends And Related Symbol Documentation	621
12.142.5.1	operator<<	621
12.143	gdcm::IconImageFilter Class Reference	621
12.143.1	Detailed Description	622
12.143.2	Constructor & Destructor Documentation	622
12.143.2.1	IconImageFilter()	622
12.143.2.2	~IconImageFilter()	622
12.143.3	Member Function Documentation	623
12.143.3.1	Extract()	623
12.143.3.2	ExtractIconImages()	623
12.143.3.3	ExtractVeproIconImages()	623
12.143.3.4	GetFile() [1/2]	623
12.143.3.5	GetFile() [2/2]	623
12.143.3.6	GetIconImage()	623
12.143.3.7	GetNumberOfIconImages()	624
12.143.3.8	SetFile()	624
12.144	gdcm::IconImageGenerator Class Reference	624
12.144.1	Detailed Description	625
12.144.2	Constructor & Destructor Documentation	625
12.144.2.1	IconImageGenerator()	625
12.144.2.2	~IconImageGenerator()	625
12.144.3	Member Function Documentation	625
12.144.3.1	AutoPixelMinMax()	625
12.144.3.2	ConvertRGBToPaletteColor()	626
12.144.3.3	Generate()	626
12.144.3.4	GetIconImage()	626
12.144.3.5	GetPixmap() [1/2]	626
12.144.3.6	GetPixmap() [2/2]	626
12.144.3.7	SetOutputDimensions()	626
12.144.3.8	SetOutsideValuePixel()	627
12.144.3.9	SetPixelMinMax()	627
12.144.3.10	SetPixmap()	627
12.145	gdcm::ignore_char Struct Reference	627
12.145.1	Constructor & Destructor Documentation	628
12.145.1.1	ignore_char()	628
12.145.2	Member Data Documentation	628
12.145.2.1	m_char	628
12.146	gdcm::Image Class Reference	628
12.146.1	Detailed Description	633

12.146.2 Constructor & Destructor Documentation	633
12.146.2.1 Image()	633
12.146.2.2 ~Image()	633
12.146.3 Member Function Documentation	634
12.146.3.1 GetDirectionCosines() [1/2]	634
12.146.3.2 GetDirectionCosines() [2/2]	634
12.146.3.3 GetIntercept()	634
12.146.3.4 GetOrigin() [1/2]	634
12.146.3.5 GetOrigin() [2/2]	634
12.146.3.6 GetSlope()	634
12.146.3.7 GetSpacing() [1/2]	634
12.146.3.8 GetSpacing() [2/2]	635
12.146.3.9 Print()	635
12.146.3.10 SetDirectionCosines() [1/3]	635
12.146.3.11 SetDirectionCosines() [2/3]	635
12.146.3.12 SetDirectionCosines() [3/3]	635
12.146.3.13 SetIntercept()	635
12.146.3.14 SetOrigin() [1/3]	636
12.146.3.15 SetOrigin() [2/3]	636
12.146.3.16 SetOrigin() [3/3]	636
12.146.3.17 SetSlope()	636
12.146.3.18 SetSpacing() [1/2]	636
12.146.3.19 SetSpacing() [2/2]	636
12.147 gdcmm::ImageApplyLookupTable Class Reference	637
12.147.1 Detailed Description	639
12.147.2 Constructor & Destructor Documentation	639
12.147.2.1 ImageApplyLookupTable()	639
12.147.2.2 ~ImageApplyLookupTable()	640
12.147.3 Member Function Documentation	640
12.147.3.1 Apply()	640
12.147.3.2 SetRGB8()	640
12.148 gdcmm::ImageChangePhotometricInterpretation Class Reference	640
12.148.1 Detailed Description	643
12.148.2 Constructor & Destructor Documentation	643
12.148.2.1 ImageChangePhotometricInterpretation()	643
12.148.2.2 ~ImageChangePhotometricInterpretation()	643
12.148.3 Member Function Documentation	643
12.148.3.1 Change()	643
12.148.3.2 ChangeMonochrome()	643
12.148.3.3 ChangeRGB2YBR()	643

12.148.3.4	ChangeYBR2RGB()	643
12.148.3.5	GetPhotometricInterpretation()	643
12.148.3.6	RGB2YBR()	644
12.148.3.7	SetPhotometricInterpretation()	644
12.148.3.8	YBR2RGB()	644
12.149	gdcm::ImageChangePlanarConfiguration Class Reference	645
12.149.1	Detailed Description	647
12.149.2	Constructor & Destructor Documentation	648
12.149.2.1	ImageChangePlanarConfiguration()	648
12.149.2.2	~ImageChangePlanarConfiguration()	648
12.149.3	Member Function Documentation	648
12.149.3.1	Change()	648
12.149.3.2	GetPlanarConfiguration()	648
12.149.3.3	RGBPixelsToRGBPlanes()	648
12.149.3.4	RGBPlanesToRGBPixels()	649
12.149.3.5	SetPlanarConfiguration()	649
12.150	gdcm::ImageChangeTransferSyntax Class Reference	649
12.150.1	Detailed Description	653
12.150.2	Constructor & Destructor Documentation	653
12.150.2.1	ImageChangeTransferSyntax()	653
12.150.2.2	~ImageChangeTransferSyntax()	653
12.150.3	Member Function Documentation	653
12.150.3.1	Change()	653
12.150.3.2	GetTransferSyntax()	654
12.150.3.3	SetCompressIconImage()	654
12.150.3.4	SetForce()	654
12.150.3.5	SetTransferSyntax()	654
12.150.3.6	SetUserCodec()	655
12.150.3.7	TryJPEG2000Codec()	655
12.150.3.8	TryJPEGCodec()	655
12.150.3.9	TryJPEGLSCodec()	655
12.150.3.10	TryRAWCodec()	655
12.150.3.11	TryRLECodec()	656
12.151	gdcm::ImageCodec Class Reference	656
12.151.1	Detailed Description	659
12.151.2	Member Typedef Documentation	659
12.151.2.1	LUTPtr	659
12.151.3	Constructor & Destructor Documentation	659
12.151.3.1	ImageCodec()	659
12.151.3.2	~ImageCodec()	659

12.151.4 Member Function Documentation	659
12.151.4.1 AppendFrameEncode()	659
12.151.4.2 AppendRowEncode()	660
12.151.4.3 CanCode()	660
12.151.4.4 CanDecode()	660
12.151.4.5 CleanupUnusedBits()	660
12.151.4.6 Clone()	660
12.151.4.7 Decode()	661
12.151.4.8 DecodeByStreams()	661
12.151.4.9 DoByteSwap()	661
12.151.4.10 DoInvertMonochrome()	661
12.151.4.11 DoOverlayCleanup()	661
12.151.4.12 DoPaddedCompositePixelCode()	661
12.151.4.13 DoPlanarConfiguration()	662
12.151.4.14 DoSimpleCopy()	662
12.151.4.15 DoYBR()	662
12.151.4.16 DoYBRFull422()	662
12.151.4.17 GetDimensions()	662
12.151.4.18 GetHeaderInfo()	662
12.151.4.19 GetLossyFlag()	662
12.151.4.20 GetLUT()	663
12.151.4.21 GetNeedByteSwap()	663
12.151.4.22 GetNumberOfDimensions()	663
12.151.4.23 GetPhotometricInterpretation()	663
12.151.4.24 GetPixelFormat() [1/2]	663
12.151.4.25 GetPixelFormat() [2/2]	663
12.151.4.26 GetPlanarConfiguration()	663
12.151.4.27 IsFrameEncoder()	664
12.151.4.28 IsLossy()	664
12.151.4.29 IsRowEncoder()	664
12.151.4.30 IsValid()	664
12.151.4.31 SetDimensions() [1/2]	664
12.151.4.32 SetDimensions() [2/2]	664
12.151.4.33 SetLossyFlag()	664
12.151.4.34 SetLUT()	665
12.151.4.35 SetNeedByteSwap()	665
12.151.4.36 SetNeedOverlayCleanup()	665
12.151.4.37 SetNumberOfDimensions()	665
12.151.4.38 SetPhotometricInterpretation()	665
12.151.4.39 SetPixelFormat()	666

12.151.4.40	SetPlanarConfiguration()	666
12.151.4.41	StartEncode()	666
12.151.4.42	StopEncode()	666
12.151.5	Friends And Related Symbol Documentation	666
12.151.5.1	FileChangeTransferSyntax	666
12.151.5.2	ImageChangePhotometricInterpretation	667
12.151.6	Member Data Documentation	667
12.151.6.1	Dimensions	667
12.151.6.2	LossyFlag	667
12.151.6.3	LUT	667
12.151.6.4	NeedByteSwap	667
12.151.6.5	NeedOverlayCleanup	667
12.151.6.6	NumberOfDimensions	667
12.151.6.7	PF	668
12.151.6.8	PI	668
12.151.6.9	PlanarConfiguration	668
12.151.6.10	RequestPaddedCompositePixelCode	668
12.151.6.11	RequestPlanarConfiguration	668
12.152	gdcm::ImageConverter Class Reference	668
12.152.1	Detailed Description	669
12.152.2	Constructor & Destructor Documentation	669
12.152.2.1	ImageConverter()	669
12.152.2.2	~ImageConverter()	669
12.152.3	Member Function Documentation	669
12.152.3.1	Convert()	669
12.152.3.2	GetOuput()	669
12.152.3.3	SetInput()	669
12.153	gdcm::ImageFragmentSplitter Class Reference	670
12.153.1	Detailed Description	672
12.153.2	Constructor & Destructor Documentation	672
12.153.2.1	ImageFragmentSplitter()	672
12.153.2.2	~ImageFragmentSplitter()	673
12.153.3	Member Function Documentation	673
12.153.3.1	GetFragmentSizeMax()	673
12.153.3.2	SetForce()	673
12.153.3.3	SetFragmentSizeMax()	673
12.153.3.4	Split()	673
12.154	gdcm::ImageHelper Class Reference	673
12.154.1	Detailed Description	675
12.154.2	Member Function Documentation	675

12.154.2.1 ComputeMediaStorageFromModality()	675
12.154.2.2 ComputeSpacingFromImagePositionPatient()	675
12.154.2.3 GetDimensionsValue()	675
12.154.2.4 GetDirectionCosinesFromDataSet()	676
12.154.2.5 GetDirectionCosinesValue()	676
12.154.2.6 GetForcePixelSpacing()	676
12.154.2.7 GetForceRescaleInterceptSlope()	676
12.154.2.8 GetLUT()	676
12.154.2.9 GetOriginValue()	676
12.154.2.10 GetPhotometricInterpretationValue()	676
12.154.2.11 GetPixelFormatValue()	677
12.154.2.12 GetPlanarConfigurationValue()	677
12.154.2.13 GetPMSRescaleInterceptSlope()	677
12.154.2.14 GetPointerFromElement()	677
12.154.2.15 GetRealWorldValueMappingContent()	677
12.154.2.16 GetRescaleInterceptSlopeValue()	677
12.154.2.17 GetSecondaryCaptureImagePlaneModule()	678
12.154.2.18 GetSpacingTagFromMediaStorage()	678
12.154.2.19 GetSpacingValue()	678
12.154.2.20 GetZSpacingTagFromMediaStorage()	678
12.154.2.21 SetDimensionsValue()	678
12.154.2.22 SetDirectionCosinesValue()	678
12.154.2.23 SetForcePixelSpacing()	678
12.154.2.24 SetForceRescaleInterceptSlope()	679
12.154.2.25 SetOriginValue()	679
12.154.2.26 SetPMSRescaleInterceptSlope()	679
12.154.2.27 SetRescaleInterceptSlopeValue()	679
12.154.2.28 SetSecondaryCaptureImagePlaneModule()	679
12.154.2.29 SetSpacingValue()	680
12.155 gdcm::ImageReader Class Reference	680
12.155.1 Detailed Description	683
12.155.2 Constructor & Destructor Documentation	683
12.155.2.1 ImageReader()	683
12.155.2.2 ~ImageReader()	683
12.155.3 Member Function Documentation	683
12.155.3.1 GetImage() [1/2]	683
12.155.3.2 GetImage() [2/2]	683
12.155.3.3 Read()	684
12.155.3.4 ReadACRNEMAIImage()	684
12.155.3.5 ReadImage()	684

12.156	gdcm::ImageRegionReader Class Reference	685
12.156.1	Detailed Description	688
12.156.2	Constructor & Destructor Documentation	688
12.156.2.1	ImageRegionReader()	688
12.156.2.2	~ImageRegionReader()	689
12.156.3	Member Function Documentation	689
12.156.3.1	ComputeBufferLength()	689
12.156.3.2	GetRegion()	689
12.156.3.3	Read()	689
12.156.3.4	ReadInformation()	689
12.156.3.5	ReadIntoBuffer()	690
12.156.3.6	SetRegion()	690
12.157	gdcm::ImageToImageFilter Class Reference	690
12.157.1	Detailed Description	692
12.157.2	Constructor & Destructor Documentation	692
12.157.2.1	ImageToImageFilter()	692
12.157.2.2	~ImageToImageFilter()	692
12.157.3	Member Function Documentation	692
12.157.3.1	GetInput()	692
12.157.3.2	GetOutput()	693
12.158	gdcm::ImageWriter Class Reference	693
12.158.1	Detailed Description	696
12.158.2	Constructor & Destructor Documentation	696
12.158.2.1	ImageWriter()	696
12.158.2.2	~ImageWriter()	696
12.158.3	Member Function Documentation	696
12.158.3.1	ComputeTargetMediaStorage()	696
12.158.3.2	GetImage() [1/2]	697
12.158.3.3	GetImage() [2/2]	697
12.158.3.4	Write()	697
12.159	gdcm::network::ImplementationClassUIDSub Class Reference	697
12.159.1	Detailed Description	698
12.159.2	Constructor & Destructor Documentation	698
12.159.2.1	ImplementationClassUIDSub()	698
12.159.3	Member Function Documentation	698
12.159.3.1	Print()	698
12.159.3.2	Read()	698
12.159.3.3	Size()	698
12.159.3.4	Write()	698
12.160	gdcm::network::ImplementationUIDSub Class Reference	699

12.160.1 Detailed Description	699
12.160.2 Constructor & Destructor Documentation	699
12.160.2.1 ImplementationUIDSub()	699
12.160.3 Member Function Documentation	699
12.160.3.1 Write()	699
12.161 gdcmm::network::ImplementationVersionNameSub Class Reference	699
12.161.1 Detailed Description	700
12.161.2 Constructor & Destructor Documentation	700
12.161.2.1 ImplementationVersionNameSub()	700
12.161.3 Member Function Documentation	700
12.161.3.1 Print()	700
12.161.3.2 Read()	700
12.161.3.3 Size()	700
12.161.3.4 Write()	700
12.162 gdcmm::ImplicitDataElement Class Reference	701
12.162.1 Detailed Description	703
12.162.2 Member Function Documentation	704
12.162.2.1 GetLength()	704
12.162.2.2 Read()	704
12.162.2.3 ReadPreValue()	704
12.162.2.4 ReadValue()	704
12.162.2.5 ReadValueWithLength()	704
12.162.2.6 ReadWithLength()	704
12.162.2.7 Write()	705
12.163 gdcmm::InitializeEvent Class Reference	705
12.164 gdcmm::IOD Class Reference	706
12.164.1 Detailed Description	707
12.164.2 Member Typedef Documentation	707
12.164.2.1 MapIODEntry	707
12.164.2.2 SizeType	707
12.164.3 Constructor & Destructor Documentation	707
12.164.3.1 IOD()	707
12.164.4 Member Function Documentation	708
12.164.4.1 AddIODEntry()	708
12.164.4.2 Clear()	708
12.164.4.3 GetIODEntry()	708
12.164.4.4 GetNumberOfIODs()	708
12.164.4.5 GetTypeFromTag()	708
12.164.5 Friends And Related Symbol Documentation	708
12.164.5.1 operator<<	708

12.165	gdcmm::IODEntry Class Reference	709
12.165.1	Detailed Description	709
12.165.2	Constructor & Destructor Documentation	710
12.165.2.1	IODEntry()	710
12.165.3	Member Function Documentation	710
12.165.3.1	GetIE()	710
12.165.3.2	GetName()	710
12.165.3.3	GetRef()	710
12.165.3.4	GetUsage()	710
12.165.3.5	GetUsageType()	710
12.165.3.6	SetIE()	710
12.165.3.7	SetName()	711
12.165.3.8	SetRef()	711
12.165.3.9	SetUsage()	711
12.165.4	Friends And Related Symbol Documentation	711
12.165.4.1	operator<<	711
12.166	gdcmm::IODs Class Reference	711
12.166.1	Detailed Description	712
12.166.2	Member Typedef Documentation	712
12.166.2.1	IODMapType	712
12.166.2.2	IODMapTypeConstIterator	712
12.166.2.3	IODName	713
12.166.3	Constructor & Destructor Documentation	713
12.166.3.1	IODs()	713
12.166.4	Member Function Documentation	713
12.166.4.1	AddIOD()	713
12.166.4.2	Begin()	713
12.166.4.3	Clear()	713
12.166.4.4	End()	713
12.166.4.5	GetIOD()	714
12.166.5	Friends And Related Symbol Documentation	714
12.166.5.1	operator<<	714
12.167	gdcmm::IPPSorter Class Reference	714
12.167.1	Detailed Description	716
12.167.2	Constructor & Destructor Documentation	717
12.167.2.1	IPPSorter()	717
12.167.3	Member Function Documentation	717
12.167.3.1	GetDirectionCosinesTolerance()	717
12.167.3.2	GetZSpacing()	717
12.167.3.3	GetZSpacingTolerance()	717

12.167.3.4 SetComputeZSpacing()	718
12.167.3.5 SetDirectionCosinesTolerance()	718
12.167.3.6 SetDropDuplicatePositions()	718
12.167.3.7 SetZSpacingTolerance()	718
12.167.3.8 Sort()	719
12.167.4 Member Data Documentation	719
12.167.4.1 ComputeZSpacing	719
12.167.4.2 DirCosTolerance	719
12.167.4.3 DropDuplicatePositions	719
12.167.4.4 ZSpacing	719
12.167.4.5 ZTolerance	720
12.168 gdcm::Item Class Reference	720
12.168.1 Detailed Description	723
12.168.2 Constructor & Destructor Documentation	723
12.168.2.1 Item() [1/2]	723
12.168.2.2 Item() [2/2]	724
12.168.3 Member Function Documentation	724
12.168.3.1 Clear()	724
12.168.3.2 FindDataElement()	724
12.168.3.3 GetDataElement()	724
12.168.3.4 GetLength()	724
12.168.3.5 GetNestedDataSet() [1/2]	724
12.168.3.6 GetNestedDataSet() [2/2]	725
12.168.3.7 InsertDataElement()	725
12.168.3.8 Read()	725
12.168.3.9 SetNestedDataSet()	725
12.168.3.10 Write()	725
12.168.4 Friends And Related Symbol Documentation	726
12.168.4.1 operator<<	726
12.169 gdcm::IterationEvent Class Reference	726
12.170 gdcm::JPEG12Codec Class Reference	727
12.170.1 Detailed Description	731
12.170.2 Constructor & Destructor Documentation	731
12.170.2.1 JPEG12Codec()	731
12.170.2.2 ~JPEG12Codec()	731
12.170.3 Member Function Documentation	731
12.170.3.1 DecodeByStreams()	731
12.170.3.2 EncodeBuffer()	732
12.170.3.3 GetHeaderInfo()	732
12.170.3.4 InternalCode()	732

12.170.3.5 IsStateSuspension()	732
12.171 gdcm::JPEG16Codec Class Reference	733
12.171.1 Detailed Description	736
12.171.2 Constructor & Destructor Documentation	736
12.171.2.1 JPEG16Codec()	736
12.171.2.2 ~JPEG16Codec()	736
12.171.3 Member Function Documentation	736
12.171.3.1 DecodeByStreams()	736
12.171.3.2 EncodeBuffer()	737
12.171.3.3 GetHeaderInfo()	737
12.171.3.4 InternalCode()	737
12.171.3.5 IsStateSuspension()	737
12.172 gdcm::JPEG2000Codec Class Reference	738
12.172.1 Detailed Description	741
12.172.2 Constructor & Destructor Documentation	741
12.172.2.1 JPEG2000Codec()	741
12.172.2.2 ~JPEG2000Codec()	741
12.172.3 Member Function Documentation	741
12.172.3.1 AppendFrameEncode()	741
12.172.3.2 AppendRowEncode()	742
12.172.3.3 CanCode()	742
12.172.3.4 CanDecode()	742
12.172.3.5 Clone()	742
12.172.3.6 Code()	742
12.172.3.7 Decode()	743
12.172.3.8 DecodeByStreams()	743
12.172.3.9 DecodeExtent()	743
12.172.3.10 GetHeaderInfo()	743
12.172.3.11 GetQuality()	743
12.172.3.12 GetRate()	744
12.172.3.13 IsFrameEncoder()	744
12.172.3.14 IsRowEncoder()	744
12.172.3.15 SetMCT()	744
12.172.3.16 SetNumberOfResolutions()	744
12.172.3.17 SetNumberOfThreadsForDecompression()	744
12.172.3.18 SetQuality()	744
12.172.3.19 SetRate()	745
12.172.3.20 SetReversible()	745
12.172.3.21 SetTileSize()	745
12.172.3.22 StartEncode()	745

12.172.3.23 StopEncode()	745
12.172.4 Friends And Related Symbol Documentation	745
12.172.4.1 Bitmap	745
12.172.4.2 ImageRegionReader	746
12.173 gdcm::JPEG8Codec Class Reference	746
12.173.1 Detailed Description	749
12.173.2 Constructor & Destructor Documentation	750
12.173.2.1 JPEG8Codec()	750
12.173.2.2 ~JPEG8Codec()	750
12.173.3 Member Function Documentation	750
12.173.3.1 DecodeByStreams()	750
12.173.3.2 EncodeBuffer()	750
12.173.3.3 GetHeaderInfo()	750
12.173.3.4 InternalCode()	750
12.173.3.5 IsStateSuspension()	751
12.174 gdcm::JPEGCodec Class Reference	751
12.174.1 Detailed Description	754
12.174.2 Constructor & Destructor Documentation	755
12.174.2.1 JPEGCodec()	755
12.174.2.2 ~JPEGCodec()	755
12.174.3 Member Function Documentation	755
12.174.3.1 AppendFrameEncode()	755
12.174.3.2 AppendRowEncode()	755
12.174.3.3 CanCode()	755
12.174.3.4 CanDecode()	756
12.174.3.5 Clone()	756
12.174.3.6 Code()	756
12.174.3.7 ComputeOffsetTable()	756
12.174.3.8 Decode()	756
12.174.3.9 DecodeByStreams()	757
12.174.3.10 DecodeExtent()	757
12.174.3.11 EncodeBuffer()	757
12.174.3.12 GetHeaderInfo()	757
12.174.3.13 GetLossless()	757
12.174.3.14 GetQuality()	758
12.174.3.15 IsFrameEncoder()	758
12.174.3.16 IsRowEncoder()	758
12.174.3.17 IsStateSuspension()	758
12.174.3.18 IsValid()	758
12.174.3.19 SetBitSample()	758

12.174.3.20 SetLossless()	758
12.174.3.21 SetPixelFormat()	759
12.174.3.22 SetQuality()	759
12.174.3.23 StartEncode()	759
12.174.3.24 StopEncode()	759
12.174.4 Friends And Related Symbol Documentation	759
12.174.4.1 ImageRegionReader	759
12.174.5 Member Data Documentation	760
12.174.5.1 BitSample	760
12.174.5.2 Quality	760
12.175 gdcm::JPEGLSCodec Class Reference	760
12.175.1 Detailed Description	763
12.175.2 Constructor & Destructor Documentation	763
12.175.2.1 JPEGLSCodec()	763
12.175.2.2 ~JPEGLSCodec()	764
12.175.3 Member Function Documentation	764
12.175.3.1 AppendFrameEncode()	764
12.175.3.2 AppendRowEncode()	764
12.175.3.3 CanCode()	764
12.175.3.4 CanDecode()	764
12.175.3.5 Clone()	765
12.175.3.6 Code()	765
12.175.3.7 Decode() [1/2]	765
12.175.3.8 Decode() [2/2]	765
12.175.3.9 DecodeExtent()	765
12.175.3.10 GetBufferLength()	766
12.175.3.11 GetHeaderInfo()	766
12.175.3.12 GetLossless()	766
12.175.3.13 IsFrameEncoder()	766
12.175.3.14 IsRowEncoder()	766
12.175.3.15 SetBufferLength()	766
12.175.3.16 SetLossless()	766
12.175.3.17 SetLossyError()	766
12.175.3.18 StartEncode()	767
12.175.3.19 StopEncode()	767
12.175.4 Friends And Related Symbol Documentation	767
12.175.4.1 ImageRegionReader	767
12.176 gdcm::JSON Class Reference	767
12.176.1 Detailed Description	768
12.176.2 Constructor & Destructor Documentation	768

12.176.2.1 JSON()	768
12.176.2.2 ~JSON()	768
12.176.3 Member Function Documentation	768
12.176.3.1 Code()	768
12.176.3.2 Decode()	768
12.176.3.3 GetPrettyPrint()	768
12.176.3.4 PrettyPrintOff()	769
12.176.3.5 PrettyPrintOn()	769
12.176.3.6 SetPrettyPrint()	769
12.177 gdcmm::KAKADUCodec Class Reference	769
12.177.1 Detailed Description	772
12.177.2 Constructor & Destructor Documentation	772
12.177.2.1 KAKADUCodec()	772
12.177.2.2 ~KAKADUCodec()	772
12.177.3 Member Function Documentation	772
12.177.3.1 CanCode()	772
12.177.3.2 CanDecode()	772
12.177.3.3 Clone()	772
12.177.3.4 Code()	773
12.177.3.5 Decode()	773
12.178 gdcmm::LO Class Reference	773
12.178.1 Detailed Description	774
12.178.2 Member Typedef Documentation	775
12.178.2.1 const_iterator	775
12.178.2.2 const_reference	775
12.178.2.3 const_reverse_iterator	775
12.178.2.4 difference_type	775
12.178.2.5 iterator	775
12.178.2.6 pointer	775
12.178.2.7 reference	775
12.178.2.8 reverse_iterator	775
12.178.2.9 size_type	775
12.178.2.10 Superclass	776
12.178.2.11 value_type	776
12.178.3 Constructor & Destructor Documentation	776
12.178.3.1 LO() [1/4]	776
12.178.3.2 LO() [2/4]	776
12.178.3.3 LO() [3/4]	776
12.178.3.4 LO() [4/4]	776
12.178.4 Member Function Documentation	776

12.178.4.1 IsValid()	776
12.179 gdcm::LookupTable Class Reference	777
12.179.1 Detailed Description	779
12.179.2 Member Enumeration Documentation	779
12.179.2.1 LookupTableType	779
12.179.3 Constructor & Destructor Documentation	779
12.179.3.1 LookupTable() [1/2]	779
12.179.3.2 ~LookupTable()	780
12.179.3.3 LookupTable() [2/2]	780
12.179.4 Member Function Documentation	780
12.179.4.1 Allocate()	780
12.179.4.2 Clear()	780
12.179.4.3 Decode() [1/2]	780
12.179.4.4 Decode() [2/2]	780
12.179.4.5 Decode8()	781
12.179.4.6 GetBitSample()	781
12.179.4.7 GetBufferAsRGBA()	781
12.179.4.8 GetLUT()	781
12.179.4.9 GetLUTDescriptor()	781
12.179.4.10 GetLUTLength()	781
12.179.4.11 GetPointer()	782
12.179.4.12 InitializeBlueLUT()	782
12.179.4.13 Initialized()	782
12.179.4.14 InitializeGreenLUT()	782
12.179.4.15 InitializeLUT()	782
12.179.4.16 InitializeRedLUT()	782
12.179.4.17 IsRGB8()	783
12.179.4.18 Print()	783
12.179.4.19 SetBlueLUT()	783
12.179.4.20 SetGreenLUT()	783
12.179.4.21 SetLUT()	783
12.179.4.22 SetRedLUT()	783
12.179.4.23 WriteBufferAsRGBA()	784
12.179.5 Member Data Documentation	784
12.179.5.1 BitSample	784
12.179.5.2 IncompleteLUT	784
12.179.5.3 Internal	784
12.180 gdcm::Scanner2::ltstr Struct Reference	784
12.180.1 Member Function Documentation	785
12.180.1.1 operator>()()	785

12.181	gdcmm::Scanner::ltstr Struct Reference	785
12.181.1	Member Function Documentation	785
12.181.1.1	operator>()()	785
12.182	gdcmm::StrictScanner2::ltstr Struct Reference	785
12.182.1	Member Function Documentation	786
12.182.1.1	operator>()()	786
12.183	gdcmm::StrictScanner::ltstr Struct Reference	786
12.183.1	Member Function Documentation	786
12.183.1.1	operator>()()	786
12.184	gdcmm::Macro Class Reference	787
12.184.1	Detailed Description	787
12.184.2	Member Typedef Documentation	787
12.184.2.1	ArrayIncludeMacrosType	787
12.184.2.2	MapModuleEntry	788
12.184.3	Constructor & Destructor Documentation	788
12.184.3.1	Macro()	788
12.184.4	Member Function Documentation	788
12.184.4.1	AddMacroEntry()	788
12.184.4.2	Clear()	788
12.184.4.3	FindMacroEntry()	788
12.184.4.4	GetMacroEntry()	788
12.184.4.5	GetName()	788
12.184.4.6	SetName()	789
12.184.4.7	Verify()	789
12.184.5	Friends And Related Symbol Documentation	789
12.184.5.1	operator<<	789
12.185	gdcmm::Macros Class Reference	789
12.185.1	Detailed Description	790
12.185.2	Member Typedef Documentation	790
12.185.2.1	ModuleMapType	790
12.185.3	Constructor & Destructor Documentation	790
12.185.3.1	Macros()	790
12.185.4	Member Function Documentation	790
12.185.4.1	AddMacro()	790
12.185.4.2	Clear()	791
12.185.4.3	GetMacro()	791
12.185.4.4	IsEmpty()	791
12.185.5	Friends And Related Symbol Documentation	791
12.185.5.1	operator<<	791
12.186	gdcmm::network::MaximumLengthSub Class Reference	791

12.186.1 Detailed Description	792
12.186.2 Constructor & Destructor Documentation	792
12.186.2.1 MaximumLengthSub()	792
12.186.3 Member Function Documentation	792
12.186.3.1 GetMaximumLength()	792
12.186.3.2 Print()	792
12.186.3.3 Read()	792
12.186.3.4 SetMaximumLength()	792
12.186.3.5 Size()	792
12.186.3.6 Write()	793
12.187 gdcm::MD5 Class Reference	793
12.187.1 Detailed Description	793
12.187.2 Member Function Documentation	793
12.187.2.1 Compute()	793
12.187.2.2 ComputeFile()	794
12.188 gdcm::MEC_MR3 Class Reference	794
12.188.1 Detailed Description	794
12.188.2 Member Function Documentation	794
12.188.2.1 GetCanonMECMR3Tag()	794
12.188.2.2 GetPMTFInformationDataTag()	794
12.188.2.3 GetToshibaMECMR3Tag()	795
12.188.2.4 Print()	795
12.189 gdcm::MediaStorage Class Reference	795
12.189.1 Detailed Description	798
12.189.2 Member Enumeration Documentation	799
12.189.2.1 MStype	799
12.189.2.2 ObjectType	801
12.189.3 Constructor & Destructor Documentation	802
12.189.3.1 MediaStorage()	802
12.189.4 Member Function Documentation	802
12.189.4.1 GetModality()	802
12.189.4.2 GetModalityDimension()	802
12.189.4.3 GetMSString()	802
12.189.4.4 GetMStype()	803
12.189.4.5 GetNumberOfModality()	803
12.189.4.6 GetNumberOfMSString()	803
12.189.4.7 GetNumberOfMStype()	803
12.189.4.8 GetString()	803
12.189.4.9 GuessFromModality()	803
12.189.4.10 IsImage()	804

12.189.4.11	IsUndefined()	804
12.189.4.12	operator MStype()	804
12.189.4.13	SetFromDataSet()	804
12.189.4.14	SetFromFile()	804
12.189.4.15	SetFromHeader()	805
12.189.4.16	SetFromModality()	805
12.189.4.17	SetFromSourceImageSequence()	805
12.189.5	Friends And Related Symbol Documentation	805
12.189.5.1	operator<<	805
12.190	gdcm::MemberCommand< T > Class Template Reference	805
12.190.1	Detailed Description	809
12.190.2	Member Typedef Documentation	809
12.190.2.1	Self	809
12.190.2.2	TConstMemberFunctionPointer	809
12.190.2.3	TMemberFunctionPointer	809
12.190.3	Constructor & Destructor Documentation	809
12.190.3.1	MemberCommand() [1/2]	809
12.190.3.2	MemberCommand() [2/2]	809
12.190.3.3	~MemberCommand()	810
12.190.4	Member Function Documentation	810
12.190.4.1	Execute() [1/2]	810
12.190.4.2	Execute() [2/2]	810
12.190.4.3	New()	810
12.190.4.4	operator=()	810
12.190.4.5	SetCallbackFunction() [1/2]	811
12.190.4.6	SetCallbackFunction() [2/2]	811
12.190.5	Member Data Documentation	811
12.190.5.1	m_ConstMemberFunction	811
12.190.5.2	m_MemberFunction	811
12.190.5.3	m_This	811
12.191	gdcm::MeshPrimitive Class Reference	812
12.191.1	Detailed Description	814
12.191.2	Member Typedef Documentation	814
12.191.2.1	PrimitivesData	814
12.191.3	Member Enumeration Documentation	814
12.191.3.1	MPType	814
12.191.4	Constructor & Destructor Documentation	815
12.191.4.1	MeshPrimitive()	815
12.191.4.2	~MeshPrimitive()	815
12.191.5	Member Function Documentation	815

12.191.5.1 AddPrimitiveData()	815
12.191.5.2 GetMPType()	815
12.191.5.3 GetMPTypeString()	815
12.191.5.4 GetNumberOfPrimitivesData()	816
12.191.5.5 GetPrimitiveData() [1/4]	816
12.191.5.6 GetPrimitiveData() [2/4]	816
12.191.5.7 GetPrimitiveData() [3/4]	816
12.191.5.8 GetPrimitiveData() [4/4]	816
12.191.5.9 GetPrimitivesData() [1/2]	816
12.191.5.10 GetPrimitivesData() [2/2]	816
12.191.5.11 GetPrimitiveType()	816
12.191.5.12 SetPrimitiveData() [1/2]	816
12.191.5.13 SetPrimitiveData() [2/2]	817
12.191.5.14 SetPrimitivesData()	817
12.191.5.15 SetPrimitiveType()	817
12.191.6 Member Data Documentation	817
12.191.6.1 PrimitiveData	817
12.191.6.2 PrimitiveType	817
12.192 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference	817
12.192.1 Detailed Description	820
12.192.2 Constructor & Destructor Documentation	820
12.192.2.1 ModalityPerformedProcedureStepCreateQuery()	820
12.192.3 Member Function Documentation	820
12.192.3.1 GetAbstractSyntaxUID()	820
12.192.3.2 GetRequiredDataSet()	820
12.192.3.3 ValidateQuery()	820
12.192.4 Friends And Related Symbol Documentation	821
12.192.4.1 QueryFactory	821
12.193 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference	821
12.193.1 Detailed Description	823
12.193.2 Constructor & Destructor Documentation	823
12.193.2.1 ModalityPerformedProcedureStepSetQuery()	823
12.193.3 Member Function Documentation	824
12.193.3.1 GetAbstractSyntaxUID()	824
12.193.3.2 GetRequiredDataSet()	824
12.193.3.3 ValidateQuery()	824
12.193.4 Friends And Related Symbol Documentation	824
12.193.4.1 QueryFactory	824
12.194 gdcm::ModifiedEvent Class Reference	825
12.195 gdcm::Module Class Reference	826

12.195.1 Detailed Description	827
12.195.2 Member Typedef Documentation	827
12.195.2.1 ArrayIncludeMacrosType	827
12.195.2.2 MapModuleEntry	827
12.195.3 Constructor & Destructor Documentation	827
12.195.3.1 Module()	827
12.195.4 Member Function Documentation	827
12.195.4.1 AddMacro()	827
12.195.4.2 AddModuleEntry()	828
12.195.4.3 Clear()	828
12.195.4.4 FindModuleEntryInMacros()	828
12.195.4.5 GetModuleEntryInMacros()	828
12.195.4.6 GetName()	828
12.195.4.7 SetName()	828
12.195.4.8 Verify()	828
12.195.5 Friends And Related Symbol Documentation	829
12.195.5.1 operator<<	829
12.196 gdcmm::ModuleEntry Class Reference	829
12.196.1 Detailed Description	831
12.196.2 Member Typedef Documentation	831
12.196.2.1 Description	831
12.196.3 Constructor & Destructor Documentation	831
12.196.3.1 ModuleEntry()	831
12.196.3.2 ~ModuleEntry()	831
12.196.4 Member Function Documentation	832
12.196.4.1 GetDescription()	832
12.196.4.2 GetName()	832
12.196.4.3 GetType()	832
12.196.4.4 SetDescription()	832
12.196.4.5 SetName()	832
12.196.4.6 SetType()	832
12.196.5 Friends And Related Symbol Documentation	833
12.196.5.1 operator<<	833
12.196.6 Member Data Documentation	833
12.196.6.1 DataElementType	833
12.196.6.2 DescriptionField	833
12.196.6.3 Name	833
12.197 gdcmm::Modules Class Reference	833
12.197.1 Detailed Description	834
12.197.2 Member Typedef Documentation	834

12.197.2.1 ModuleMapType	834
12.197.3 Constructor & Destructor Documentation	834
12.197.3.1 Modules()	834
12.197.4 Member Function Documentation	835
12.197.4.1 AddModule()	835
12.197.4.2 Clear()	835
12.197.4.3 GetModule()	835
12.197.4.4 IsEmpty()	835
12.197.5 Friends And Related Symbol Documentation	835
12.197.5.1 operator<<	835
12.198 gdcm::MovePatientRootQuery Class Reference	836
12.198.1 Detailed Description	838
12.198.2 Constructor & Destructor Documentation	838
12.198.2.1 MovePatientRootQuery()	838
12.198.3 Member Function Documentation	838
12.198.3.1 GetAbstractSyntaxUID()	838
12.198.3.2 GetTagListByLevel()	839
12.198.3.3 InitializeDataSet()	839
12.198.3.4 ValidateQuery()	839
12.198.4 Friends And Related Symbol Documentation	839
12.198.4.1 QueryFactory	839
12.199 gdcm::MoveStudyRootQuery Class Reference	840
12.199.1 Detailed Description	842
12.199.2 Constructor & Destructor Documentation	842
12.199.2.1 MoveStudyRootQuery()	842
12.199.3 Member Function Documentation	842
12.199.3.1 GetAbstractSyntaxUID()	842
12.199.3.2 GetTagListByLevel()	843
12.199.3.3 InitializeDataSet()	843
12.199.3.4 ValidateQuery()	843
12.199.4 Friends And Related Symbol Documentation	843
12.199.4.1 QueryFactory	843
12.200 gdcm::MrProtocol Class Reference	844
12.200.1 Detailed Description	844
12.200.2 Constructor & Destructor Documentation	844
12.200.2.1 MrProtocol()	844
12.200.2.2 ~MrProtocol()	845
12.200.3 Member Function Documentation	845
12.200.3.1 FindMrProtocolByName()	845
12.200.3.2 GetMrProtocolByName()	845

12.200.3.3	GetSliceArray()	845
12.200.3.4	GetVersion()	845
12.200.3.5	Load()	845
12.200.3.6	Print()	845
12.200.4	Friends And Related Symbol Documentation	846
12.200.4.1	operator<<	846
12.201	gdcmm::network::NActionRQ Class Reference	846
12.201.1	Detailed Description	847
12.201.2	Member Function Documentation	847
12.201.2.1	ConstructPDV()	847
12.202	gdcmm::network::NActionRSP Class Reference	847
12.202.1	Detailed Description	848
12.202.2	Member Function Documentation	848
12.202.2.1	ConstructPDVByDataSet()	848
12.203	gdcmm::network::NCreateRQ Class Reference	849
12.203.1	Detailed Description	850
12.203.2	Member Function Documentation	850
12.203.2.1	ConstructPDV()	850
12.204	gdcmm::network::NCreateRSP Class Reference	850
12.204.1	Detailed Description	851
12.204.2	Member Function Documentation	851
12.204.2.1	ConstructPDVByDataSet()	851
12.205	gdcmm::network::NDeleteRQ Class Reference	852
12.205.1	Detailed Description	853
12.205.2	Member Function Documentation	853
12.205.2.1	ConstructPDV()	853
12.206	gdcmm::network::NDeleteRSP Class Reference	853
12.206.1	Detailed Description	854
12.206.2	Member Function Documentation	854
12.206.2.1	ConstructPDVByDataSet()	854
12.207	gdcmm::NestedModuleEntries Class Reference	855
12.207.1	Detailed Description	856
12.207.2	Member Typedef Documentation	857
12.207.2.1	SizeType	857
12.207.3	Constructor & Destructor Documentation	857
12.207.3.1	NestedModuleEntries()	857
12.207.4	Member Function Documentation	857
12.207.4.1	AddModuleEntry()	857
12.207.4.2	GetModuleEntry() [1/2]	857
12.207.4.3	GetModuleEntry() [2/2]	857

12.207.4.4	GetNumberOfModuleEntries()	857
12.207.5	Friends And Related Symbol Documentation	858
12.207.5.1	operator<<	858
12.208	gdcmm::network::NEventReportRQ Class Reference	858
12.208.1	Detailed Description	859
12.208.2	Member Function Documentation	859
12.208.2.1	ConstructPDV()	859
12.209	gdcmm::network::NEventReportRSP Class Reference	859
12.209.1	Detailed Description	860
12.209.2	Member Function Documentation	860
12.209.2.1	ConstructPDVByDataSet()	860
12.210	gdcmm::network::NGetRQ Class Reference	861
12.210.1	Detailed Description	862
12.210.2	Member Function Documentation	862
12.210.2.1	ConstructPDV()	862
12.211	gdcmm::network::NGetRSP Class Reference	862
12.211.1	Detailed Description	863
12.211.2	Member Function Documentation	863
12.211.2.1	ConstructPDVByDataSet()	863
12.212	gdcmm::NoEvent Class Reference	864
12.212.1	Detailed Description	865
12.213	gdcmm::network::NormalizedMessageFactory Class Reference	865
12.213.1	Member Function Documentation	865
12.213.1.1	ConstructNAction()	865
12.213.1.2	ConstructNCreate()	865
12.213.1.3	ConstructNDelete()	866
12.213.1.4	ConstructNEventReport()	866
12.213.1.5	ConstructNGet()	866
12.213.1.6	ConstructNSet()	866
12.214	gdcmm::NormalizedNetworkFunctions Class Reference	866
12.214.1	Detailed Description	867
12.214.2	Member Function Documentation	867
12.214.2.1	ConstructQuery()	867
12.214.2.2	NAction()	867
12.214.2.3	NCreate()	868
12.214.2.4	NDelete()	868
12.214.2.5	NEventReport()	868
12.214.2.6	NGet()	868
12.214.2.7	NSet()	868
12.215	gdcmm::network::NSetRQ Class Reference	869

12.215.1 Detailed Description	870
12.215.2 Member Function Documentation	870
12.215.2.1 ConstructPDV()	870
12.216 gdcm::network::NSetRSP Class Reference	870
12.216.1 Detailed Description	871
12.216.2 Member Function Documentation	871
12.216.2.1 ConstructPDVByDataSet()	871
12.217 gdcm::Object Class Reference	872
12.217.1 Detailed Description	873
12.217.2 Constructor & Destructor Documentation	873
12.217.2.1 Object() [1/2]	873
12.217.2.2 ~Object()	873
12.217.2.3 Object() [2/2]	873
12.217.3 Member Function Documentation	874
12.217.3.1 operator=()	874
12.217.3.2 Print()	874
12.217.3.3 Register()	874
12.217.3.4 UnRegister()	874
12.217.4 Friends And Related Symbol Documentation	874
12.217.4.1 operator<<	874
12.217.4.2 SmartPointer	875
12.218 gdcm::OpenSSLCryptoFactory Class Reference	875
12.218.1 Constructor & Destructor Documentation	876
12.218.1.1 OpenSSLCryptoFactory()	876
12.218.2 Member Function Documentation	877
12.218.2.1 CreateCMSProvider()	877
12.218.2.2 InitOpenSSL()	877
12.219 gdcm::OpenSSLCryptographicMessageSyntax Class Reference	877
12.219.1 Constructor & Destructor Documentation	879
12.219.1.1 OpenSSLCryptographicMessageSyntax()	879
12.219.1.2 ~OpenSSLCryptographicMessageSyntax()	879
12.219.2 Member Function Documentation	879
12.219.2.1 Decrypt()	879
12.219.2.2 Encrypt()	879
12.219.2.3 GetCipherType()	879
12.219.2.4 ParseCertificateFile()	880
12.219.2.5 ParseKeyFile()	880
12.219.2.6 SetCipherType()	880
12.219.2.7 SetPassword()	880
12.220 gdcm::OpenSSLP7CryptoFactory Class Reference	881

12.220.1 Constructor & Destructor Documentation	882
12.220.1.1 OpenSSLP7CryptoFactory()	882
12.220.2 Member Function Documentation	882
12.220.2.1 CreateCMSProvider()	882
12.221 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference	883
12.221.1 Detailed Description	884
12.221.2 Constructor & Destructor Documentation	884
12.221.2.1 OpenSSLP7CryptographicMessageSyntax()	884
12.221.2.2 ~OpenSSLP7CryptographicMessageSyntax()	884
12.221.3 Member Function Documentation	884
12.221.3.1 Decrypt()	884
12.221.3.2 Encrypt()	885
12.221.3.3 GetCipherType()	885
12.221.3.4 ParseCertificateFile()	885
12.221.3.5 ParseKeyFile()	885
12.221.3.6 SetCipherType()	885
12.221.3.7 SetPassword()	886
12.222 gdcm::Orientation Class Reference	886
12.222.1 Detailed Description	887
12.222.2 Member Enumeration Documentation	887
12.222.2.1 OrientationType	887
12.222.3 Constructor & Destructor Documentation	887
12.222.3.1 Orientation()	887
12.222.3.2 ~Orientation()	887
12.222.4 Member Function Documentation	888
12.222.4.1 GetLabel()	888
12.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()	888
12.222.4.3 GetObliquityThresholdCosineValue()	888
12.222.4.4 GetType()	888
12.222.4.5 Print()	888
12.222.4.6 SetObliquityThresholdCosineValue()	889
12.222.5 Friends And Related Symbol Documentation	889
12.222.5.1 operator<<	889
12.223 gdcm::Overlay Class Reference	889
12.223.1 Detailed Description	892
12.223.2 Member Enumeration Documentation	892
12.223.2.1 OverlayType	892
12.223.3 Constructor & Destructor Documentation	892
12.223.3.1 Overlay() [1/2]	892
12.223.3.2 ~Overlay()	893

12.223.3.3 Overlay() [2/2]	893
12.223.4 Member Function Documentation	893
12.223.4.1 Decompress()	893
12.223.4.2 GetBitPosition()	893
12.223.4.3 GetBitsAllocated()	893
12.223.4.4 GetColumns()	893
12.223.4.5 GetDescription()	893
12.223.4.6 GetGroup()	894
12.223.4.7 GetOrigin()	894
12.223.4.8 GetOverlayData()	894
12.223.4.9 GetOverlayTypeAsString()	894
12.223.4.10 GetOverlayTypeFromString()	894
12.223.4.11 GetRows()	894
12.223.4.12 GetType()	894
12.223.4.13 GetTypeAsEnum()	894
12.223.4.14 GetUnpackBuffer()	895
12.223.4.15 GetUnpackBufferLength()	895
12.223.4.16 GrabOverlayFromPixelData()	895
12.223.4.17 IsEmpty()	895
12.223.4.18 IsInPixelData() [1/2]	895
12.223.4.19 IsInPixelData() [2/2]	895
12.223.4.20 IsZero()	896
12.223.4.21 operator=()	896
12.223.4.22 Print()	896
12.223.4.23 SetBitPosition()	896
12.223.4.24 SetBitsAllocated()	896
12.223.4.25 SetColumns()	896
12.223.4.26 SetDescription()	897
12.223.4.27 setFrameOrigin()	897
12.223.4.28 SetGroup()	897
12.223.4.29 SetNumberOfFrames()	897
12.223.4.30 SetOrigin()	897
12.223.4.31 SetOverlay()	897
12.223.4.32 SetRows()	898
12.223.4.33 SetType()	898
12.223.4.34 Update()	898
12.224 gdcmm::ParseException Class Reference	898
12.224.1 Detailed Description	899
12.224.2 Constructor & Destructor Documentation	899
12.224.2.1 ParseException() [1/2]	899

12.224.2.2 ~ParseException()	900
12.224.2.3 ParseException() [2/2]	900
12.224.3 Member Function Documentation	900
12.224.3.1 GetLastElement()	900
12.224.3.2 operator=()	900
12.224.3.3 SetLastElement()	900
12.225 gdcm::Parser Class Reference	900
12.225.1 Detailed Description	901
12.225.2 Member Typedef Documentation	902
12.225.2.1 EndElementHandler	902
12.225.2.2 StartElementHandler	902
12.225.3 Member Enumeration Documentation	902
12.225.3.1 ErrorType	902
12.225.4 Constructor & Destructor Documentation	902
12.225.4.1 Parser()	902
12.225.4.2 ~Parser()	902
12.225.5 Member Function Documentation	903
12.225.5.1 GetBuffer()	903
12.225.5.2 GetCurrentByteIndex()	903
12.225.5.3 GetErrorCode()	903
12.225.5.4 GetErrorString()	903
12.225.5.5 GetUserData()	903
12.225.5.6 Parse()	903
12.225.5.7 ParseBuffer()	903
12.225.5.8 Process()	903
12.225.5.9 SetElementHandler()	904
12.225.5.10 SetUserData()	904
12.226 gdcm::Patient Class Reference	904
12.226.1 Detailed Description	904
12.226.2 Constructor & Destructor Documentation	904
12.226.2.1 Patient()	904
12.227 gdcm::network::PDataTFPDU Class Reference	905
12.227.1 Detailed Description	906
12.227.2 Member Typedef Documentation	906
12.227.2.1 SizeType	906
12.227.3 Constructor & Destructor Documentation	906
12.227.3.1 PDataTFPDU()	906
12.227.4 Member Function Documentation	906
12.227.4.1 AddPresentationDataValue()	906
12.227.4.2 GetNumberOfPresentationDataValues()	906

12.227.4.3	GetPresentationDataValue()	907
12.227.4.4	IsLastFragment()	907
12.227.4.5	Print()	907
12.227.4.6	Read()	907
12.227.4.7	ReadInto()	907
12.227.4.8	Size()	907
12.227.4.9	Write()	908
12.228	gdcm::PDBElement Class Reference	908
12.228.1	Detailed Description	909
12.228.2	Constructor & Destructor Documentation	909
12.228.2.1	PDBElement()	909
12.228.3	Member Function Documentation	909
12.228.3.1	GetName()	909
12.228.3.2	GetValue()	909
12.228.3.3	operator==(())	910
12.228.3.4	SetName()	910
12.228.3.5	SetValue()	910
12.228.4	Friends And Related Symbol Documentation	910
12.228.4.1	operator<<	910
12.228.5	Member Data Documentation	910
12.228.5.1	NameField	910
12.228.5.2	ValueField	911
12.229	gdcm::PDBHeader Class Reference	911
12.229.1	Detailed Description	912
12.229.2	Constructor & Destructor Documentation	912
12.229.2.1	PDBHeader()	912
12.229.2.2	~PDBHeader()	912
12.229.3	Member Function Documentation	912
12.229.3.1	FindPDBElementByName()	912
12.229.3.2	GetPDBEEnd()	912
12.229.3.3	GetPDBElementByName()	913
12.229.3.4	GetPDBInfoTag()	913
12.229.3.5	LoadFromDataElement()	913
12.229.3.6	Print()	913
12.229.4	Friends And Related Symbol Documentation	913
12.229.4.1	operator<<	913
12.230	gdcm::PDFCodec Class Reference	914
12.230.1	Detailed Description	915
12.230.2	Constructor & Destructor Documentation	915
12.230.2.1	PDFCodec()	915

12.230.2.2 ~PDFCodec()	916
12.230.3 Member Function Documentation	916
12.230.3.1 CanCode()	916
12.230.3.2 CanDecode()	916
12.230.3.3 Decode()	916
12.231 gdcmm::network::PDUFactory Class Reference	916
12.231.1 Detailed Description	917
12.231.2 Member Function Documentation	917
12.231.2.1 ConstructAbortPDU()	917
12.231.2.2 ConstructPDU()	917
12.231.2.3 ConstructReleasePDU()	918
12.231.2.4 CreateCEchoPDU()	918
12.231.2.5 CreateCFindPDU()	918
12.231.2.6 CreateCMovePDU()	918
12.231.2.7 CreateCStoreRQPDU()	918
12.231.2.8 CreateCStoreRSPPDU()	918
12.231.2.9 CreateNActionPDU()	918
12.231.2.10 CreateNCreatePDU()	919
12.231.2.11 CreateNDeletePDU()	919
12.231.2.12 CreateNEventReportPDU()	919
12.231.2.13 CreateNGetPDU()	919
12.231.2.14 CreateNSetPDU()	919
12.231.2.15 DetermineEventByPDU()	919
12.231.2.16 GetPDVs()	919
12.232 gdcmm::PersonName Class Reference	920
12.232.1 Detailed Description	920
12.232.2 Member Function Documentation	920
12.232.2.1 GetMaxLength()	920
12.232.2.2 GetNumberOfComponents()	920
12.232.2.3 Print()	921
12.232.2.4 SetBlob()	921
12.232.2.5 SetComponents() [1/2]	921
12.232.2.6 SetComponents() [2/2]	921
12.232.3 Member Data Documentation	921
12.232.3.1 Component	921
12.232.3.2 MaxLength	921
12.232.3.3 MaxNumberOfComponents	922
12.232.3.4 Padding	922
12.232.3.5 Separator	922
12.233 gdcmm::PGXCodec Class Reference	922

12.233.1 Detailed Description	925
12.233.2 Constructor & Destructor Documentation	925
12.233.2.1 PGXCodec()	925
12.233.2.2 ~PGXCodec()	925
12.233.3 Member Function Documentation	925
12.233.3.1 CanCode()	925
12.233.3.2 CanDecode()	925
12.233.3.3 Clone()	925
12.233.3.4 GetHeaderInfo()	926
12.233.3.5 Read()	926
12.233.3.6 Write()	926
12.234 gdcmm::PhotometricInterpretation Class Reference	926
12.234.1 Detailed Description	927
12.234.2 Member Enumeration Documentation	927
12.234.2.1 PType	927
12.234.3 Constructor & Destructor Documentation	928
12.234.3.1 PhotometricInterpretation()	928
12.234.4 Member Function Documentation	928
12.234.4.1 GetPString()	928
12.234.4.2 GetPType()	928
12.234.4.3 GetSamplesPerPixel()	929
12.234.4.4 GetString()	929
12.234.4.5 GetType()	929
12.234.4.6 IsLossless()	929
12.234.4.7 IsLossy()	929
12.234.4.8 IsRetired()	929
12.234.4.9 IsSameColorSpace()	929
12.234.4.10 operator PType()	929
12.234.5 Friends And Related Symbol Documentation	930
12.234.5.1 operator<<	930
12.235 gdcmm::PixelFormat Class Reference	930
12.235.1 Detailed Description	932
12.235.2 Member Enumeration Documentation	932
12.235.2.1 ScalarType	932
12.235.3 Constructor & Destructor Documentation	933
12.235.3.1 PixelFormat() [1/3]	933
12.235.3.2 PixelFormat() [2/3]	933
12.235.3.3 PixelFormat() [3/3]	933
12.235.4 Member Function Documentation	933
12.235.4.1 GetBitsAllocated()	933

12.235.4.2	GetBitsStored()	934
12.235.4.3	GetHighBit()	934
12.235.4.4	GetMax()	934
12.235.4.5	GetMin()	934
12.235.4.6	GetPixelRepresentation()	934
12.235.4.7	GetPixelSize()	935
12.235.4.8	GetSamplesPerPixel()	935
12.235.4.9	GetScalarType()	935
12.235.4.10	GetScalarTypeAsString()	935
12.235.4.11	IsCompatible()	936
12.235.4.12	IsValid()	936
12.235.4.13	operator ScalarType()	936
12.235.4.14	operator"!=() [1/2]	936
12.235.4.15	operator"!=() [2/2]	936
12.235.4.16	operator==([1/2]	936
12.235.4.17	operator==([2/2]	936
12.235.4.18	Print()	937
12.235.4.19	SetBitsAllocated()	937
12.235.4.20	SetBitsStored()	937
12.235.4.21	SetHighBit()	937
12.235.4.22	SetPixelRepresentation()	937
12.235.4.23	SetSamplesPerPixel()	937
12.235.4.24	SetScalarType()	938
12.235.4.25	Validate()	938
12.235.5	Friends And Related Symbol Documentation	938
12.235.5.1	Bitmap	938
12.235.5.2	operator<<	938
12.236	gdcm::Pixmap Class Reference	939
12.236.1	Detailed Description	942
12.236.2	Constructor & Destructor Documentation	943
12.236.2.1	Pixmap()	943
12.236.2.2	~Pixmap()	943
12.236.3	Member Function Documentation	943
12.236.3.1	AreOverlaysInPixelData()	943
12.236.3.2	GetCurve() [1/2]	943
12.236.3.3	GetCurve() [2/2]	943
12.236.3.4	GetIconImage() [1/2]	943
12.236.3.5	GetIconImage() [2/2]	944
12.236.3.6	GetNumberOfCurves()	944
12.236.3.7	GetNumberOfOverlays()	944

12.236.3.8	GetOverlay() [1/2]	944
12.236.3.9	GetOverlay() [2/2]	944
12.236.3.10	Print()	944
12.236.3.11	RemoveOverlay()	945
12.236.3.12	SetIconImage()	945
12.236.3.13	SetNumberOfCurves()	945
12.236.3.14	SetNumberOfOverlays()	945
12.236.3.15	UnusedBitsPresentInPixelData()	945
12.236.4	Member Data Documentation	945
12.236.4.1	Curves	945
12.236.4.2	Icon	946
12.236.4.3	Overlays	946
12.237	gdcm::PixmapReader Class Reference	946
12.237.1	Detailed Description	948
12.237.2	Constructor & Destructor Documentation	949
12.237.2.1	PixmapReader()	949
12.237.2.2	~PixmapReader()	949
12.237.3	Member Function Documentation	949
12.237.3.1	GetPixmap() [1/2]	949
12.237.3.2	GetPixmap() [2/2]	949
12.237.3.3	Read()	949
12.237.3.4	ReadACRNEMAIImage()	950
12.237.3.5	ReadImage()	950
12.237.3.6	ReadImageInternal()	950
12.237.4	Member Data Documentation	950
12.237.4.1	PixelData	950
12.238	gdcm::PixmapToPixmapFilter Class Reference	950
12.238.1	Detailed Description	952
12.238.2	Constructor & Destructor Documentation	952
12.238.2.1	PixmapToPixmapFilter()	952
12.238.2.2	~PixmapToPixmapFilter()	952
12.238.3	Member Function Documentation	952
12.238.3.1	GetInput()	952
12.238.3.2	GetOutput()	952
12.238.3.3	GetOutputAsPixmap()	953
12.239	gdcm::PixmapWriter Class Reference	953
12.239.1	Detailed Description	955
12.239.2	Constructor & Destructor Documentation	956
12.239.2.1	PixmapWriter()	956
12.239.2.2	~PixmapWriter()	956

12.239.3 Member Function Documentation	956
12.239.3.1 DoIconImage()	956
12.239.3.2 GetImage() [1/2]	956
12.239.3.3 GetImage() [2/2]	956
12.239.3.4 GetPixmap() [1/2]	956
12.239.3.5 GetPixmap() [2/2]	957
12.239.3.6 PrepareWrite()	957
12.239.3.7 SetImage()	957
12.239.3.8 SetPixmap()	957
12.239.3.9 Write()	957
12.239.4 Member Data Documentation	958
12.239.4.1 PixelData	958
12.240 gdcm::PNMCodec Class Reference	958
12.240.1 Detailed Description	961
12.240.2 Constructor & Destructor Documentation	961
12.240.2.1 PNMCodec()	961
12.240.2.2 ~PNMCodec()	961
12.240.3 Member Function Documentation	961
12.240.3.1 CanCode()	961
12.240.3.2 CanDecode()	962
12.240.3.3 Clone()	962
12.240.3.4 GetBufferLength()	962
12.240.3.5 GetHeaderInfo()	962
12.240.3.6 Read()	962
12.240.3.7 SetBufferLength()	962
12.240.3.8 Write()	963
12.241 gdcm::Preamble Class Reference	963
12.241.1 Detailed Description	964
12.241.2 Constructor & Destructor Documentation	964
12.241.2.1 Preamble() [1/2]	964
12.241.2.2 ~Preamble()	964
12.241.2.3 Preamble() [2/2]	964
12.241.3 Member Function Documentation	964
12.241.3.1 Clear()	964
12.241.3.2 Create()	965
12.241.3.3 GetInternal()	965
12.241.3.4 GetLength()	965
12.241.3.5 IsEmpty()	965
12.241.3.6 IsValid()	965
12.241.3.7 operator=()	965

12.241.3.8	Print()	965
12.241.3.9	Read()	966
12.241.3.10	Remove()	966
12.241.3.11	Valid()	966
12.241.3.12	Write()	966
12.241.4	Friends And Related Symbol Documentation	966
12.241.4.1	operator<<	966
12.242	gdcmm::PresentationContext Class Reference	967
12.242.1	Detailed Description	968
12.242.2	Member Typedef Documentation	968
12.242.2.1	SizeType	968
12.242.2.2	TransferSyntaxArrayType	968
12.242.3	Constructor & Destructor Documentation	968
12.242.3.1	PresentationContext() [1/2]	968
12.242.3.2	PresentationContext() [2/2]	968
12.242.4	Member Function Documentation	969
12.242.4.1	AddTransferSyntax()	969
12.242.4.2	GetAbstractSyntax()	969
12.242.4.3	GetNumberOfTransferSyntaxes()	969
12.242.4.4	GetPresentationContextID()	969
12.242.4.5	GetTransferSyntax()	969
12.242.4.6	operator==()	969
12.242.4.7	Print()	969
12.242.4.8	SetAbstractSyntax()	970
12.242.4.9	SetPresentationContextID()	970
12.242.5	Member Data Documentation	970
12.242.5.1	AbstractSyntax	970
12.242.5.2	ID	970
12.242.5.3	TransferSyntaxes	970
12.243	gdcmm::network::PresentationContextAC Class Reference	970
12.243.1	Detailed Description	971
12.243.2	Constructor & Destructor Documentation	971
12.243.2.1	PresentationContextAC()	971
12.243.3	Member Function Documentation	971
12.243.3.1	GetPresentationContextID()	971
12.243.3.2	GetReason()	971
12.243.3.3	GetTransferSyntax()	971
12.243.3.4	Print()	972
12.243.3.5	Read()	972
12.243.3.6	SetPresentationContextID()	972

12.243.3.7 SetReason()	972
12.243.3.8 SetTransferSyntax()	972
12.243.3.9 Size()	972
12.243.3.10 Write()	972
12.244 gdcmm::PresentationContextGenerator Class Reference	972
12.244.1 Detailed Description	973
12.244.2 Member Typedef Documentation	974
12.244.2.1 PresentationContextArrayType	974
12.244.2.2 SizeType	974
12.244.3 Constructor & Destructor Documentation	974
12.244.3.1 PresentationContextGenerator()	974
12.244.4 Member Function Documentation	974
12.244.4.1 AddFromFile()	974
12.244.4.2 AddPresentationContext()	974
12.244.4.3 GenerateFromFilenames()	974
12.244.4.4 GenerateFromUID()	975
12.244.4.5 GetDefaultTransferSyntax()	975
12.244.4.6 GetPresentationContexts()	975
12.244.4.7 SetDefaultTransferSyntax()	975
12.244.4.8 SetMergeModeToAbstractSyntax()	975
12.244.4.9 SetMergeModeToTransferSyntax()	975
12.245 gdcmm::network::PresentationContextRQ Class Reference	975
12.245.1 Detailed Description	976
12.245.2 Member Typedef Documentation	976
12.245.2.1 SizeType	976
12.245.3 Constructor & Destructor Documentation	976
12.245.3.1 PresentationContextRQ() [1/3]	976
12.245.3.2 PresentationContextRQ() [2/3]	977
12.245.3.3 PresentationContextRQ() [3/3]	977
12.245.4 Member Function Documentation	977
12.245.4.1 AddTransferSyntax()	977
12.245.4.2 GetAbstractSyntax() [1/2]	977
12.245.4.3 GetAbstractSyntax() [2/2]	977
12.245.4.4 GetNumberOfTransferSyntaxes()	977
12.245.4.5 GetPresentationContextID()	977
12.245.4.6 GetTransferSyntax() [1/2]	977
12.245.4.7 GetTransferSyntax() [2/2]	978
12.245.4.8 GetTransferSyntaxes()	978
12.245.4.9 operator==()	978
12.245.4.10 Print()	978

12.245.4.11 Read()	978
12.245.4.12 SetAbstractSyntax()	978
12.245.4.13 SetPresentationContextID()	978
12.245.4.14 Size()	978
12.245.4.15 Write()	979
12.246 gdcmm::network::PresentationDataValue Class Reference	979
12.246.1 Detailed Description	979
12.246.2 Constructor & Destructor Documentation	980
12.246.2.1 PresentationDataValue()	980
12.246.3 Member Function Documentation	980
12.246.3.1 ConcatenatePDVBlobs()	980
12.246.3.2 ConcatenatePDVBlobsAsExplicit()	980
12.246.3.3 GetBlob()	980
12.246.3.4 GetIsCommand()	980
12.246.3.5 GetIsLastFragment()	980
12.246.3.6 GetMessageHeader()	980
12.246.3.7 GetPresentationContextID()	981
12.246.3.8 Print()	981
12.246.3.9 Read()	981
12.246.3.10 ReadInto()	981
12.246.3.11 SetBlob()	981
12.246.3.12 SetCommand()	981
12.246.3.13 SetDataSet()	981
12.246.3.14 SetLastFragment()	982
12.246.3.15 SetMessageHeader()	982
12.246.3.16 SetPresentationContextID()	982
12.246.3.17 Size()	982
12.246.3.18 Write()	982
12.247 gdcmm::Printer Class Reference	983
12.247.1 Detailed Description	984
12.247.2 Member Enumeration Documentation	985
12.247.2.1 PrintStyles	985
12.247.3 Constructor & Destructor Documentation	985
12.247.3.1 Printer()	985
12.247.3.2 ~Printer()	985
12.247.4 Member Function Documentation	985
12.247.4.1 GetPrintStyle()	985
12.247.4.2 Print()	985
12.247.4.3 PrintDataElement()	986
12.247.4.4 PrintDataSet()	986

12.247.4.5 PrintSQ()	986
12.247.4.6 SetColor()	986
12.247.4.7 SetFile()	986
12.247.4.8 SetStyle()	987
12.247.5 Member Data Documentation	987
12.247.5.1 F	987
12.247.5.2 MaxPrintLength	987
12.247.5.3 PrintStyle	987
12.248 gdcm::PrivateDict Class Reference	987
12.248.1 Detailed Description	988
12.248.2 Constructor & Destructor Documentation	988
12.248.2.1 PrivateDict()	988
12.248.2.2 ~PrivateDict()	988
12.248.3 Member Function Documentation	988
12.248.3.1 AddDictEntry()	988
12.248.3.2 FindDictEntry()	988
12.248.3.3 GetDictEntry()	989
12.248.3.4 IsEmpty()	989
12.248.3.5 LoadDefault()	989
12.248.3.6 PrintXML()	989
12.248.3.7 RemoveDictEntry()	989
12.248.4 Friends And Related Symbol Documentation	989
12.248.4.1 Dicts	989
12.248.4.2 operator<<	990
12.249 gdcm::PrivateTag Class Reference	990
12.249.1 Detailed Description	992
12.249.2 Constructor & Destructor Documentation	993
12.249.2.1 PrivateTag() [1/2]	993
12.249.2.2 PrivateTag() [2/2]	993
12.249.3 Member Function Documentation	993
12.249.3.1 GetAsDataElement()	993
12.249.3.2 GetOwner()	993
12.249.3.3 operator"!=() [1/2]	993
12.249.3.4 operator"!=() [2/2]	994
12.249.3.5 operator<()	994
12.249.3.6 operator=()	994
12.249.3.7 operator==() [1/2]	994
12.249.3.8 operator==() [2/2]	994
12.249.3.9 ReadFromCommaSeparatedString()	994
12.249.3.10 SetOwner()	995

12.249.4 Friends And Related Symbol Documentation	995
12.249.4.1 operator<<	995
12.250 gdcmm::ProgressEvent Class Reference	995
12.250.1 Detailed Description	997
12.250.2 Member Typedef Documentation	997
12.250.2.1 Self	997
12.250.2.2 Superclass	997
12.250.3 Constructor & Destructor Documentation	997
12.250.3.1 ProgressEvent() [1/2]	997
12.250.3.2 ~ProgressEvent()	997
12.250.3.3 ProgressEvent() [2/2]	997
12.250.4 Member Function Documentation	998
12.250.4.1 CheckEvent()	998
12.250.4.2 GetEventName()	998
12.250.4.3 GetProgress()	998
12.250.4.4 MakeObject()	998
12.250.4.5 operator=()	998
12.250.4.6 SetProgress()	998
12.251 gdcmm::PVRGCodec Class Reference	999
12.251.1 Detailed Description	1002
12.251.2 Constructor & Destructor Documentation	1002
12.251.2.1 PVRGCodec()	1002
12.251.2.2 ~PVRGCodec()	1002
12.251.3 Member Function Documentation	1002
12.251.3.1 CanCode()	1002
12.251.3.2 CanDecode()	1002
12.251.3.3 Clone()	1003
12.251.3.4 Code()	1003
12.251.3.5 Decode()	1003
12.251.3.6 SetLossyFlag()	1003
12.252 gdcmm::PythonFilter Class Reference	1003
12.252.1 Detailed Description	1004
12.252.2 Constructor & Destructor Documentation	1004
12.252.2.1 PythonFilter()	1004
12.252.2.2 ~PythonFilter()	1004
12.252.3 Member Function Documentation	1004
12.252.3.1 GetFile() [1/2]	1004
12.252.3.2 GetFile() [2/2]	1004
12.252.3.3 SetDicts()	1004
12.252.3.4 SetFile()	1005

12.252.3.5 ToPyObject()	1005
12.252.3.6 UseDictAlways()	1005
12.253 gdcm::QueryBase Class Reference	1005
12.253.1 Detailed Description	1006
12.253.2 Constructor & Destructor Documentation	1006
12.253.2.1 ~QueryBase()	1006
12.253.3 Member Function Documentation	1006
12.253.3.1 GetAllRequiredTags()	1006
12.253.3.2 GetAllTags()	1006
12.253.3.3 GetHierarchicalSearchTags()	1007
12.253.3.4 GetName()	1007
12.253.3.5 GetOptionalTags()	1007
12.253.3.6 GetQueryLevel()	1007
12.253.3.7 GetRequiredTags()	1007
12.253.3.8 GetUniqueTags()	1007
12.254 gdcm::QueryFactory Class Reference	1008
12.254.1 Detailed Description	1008
12.254.2 Member Function Documentation	1008
12.254.2.1 GetCharacterFromCurrentLocale()	1008
12.254.2.2 ListCharSets()	1008
12.254.2.3 ProduceCharacterSetDataElement()	1009
12.254.2.4 ProduceQuery() [1/2]	1009
12.254.2.5 ProduceQuery() [2/2]	1009
12.255 gdcm::QueryImage Class Reference	1009
12.255.1 Detailed Description	1010
12.255.2 Member Function Documentation	1010
12.255.2.1 GetHierarchicalSearchTags()	1010
12.255.2.2 GetName()	1011
12.255.2.3 GetOptionalTags()	1011
12.255.2.4 GetQueryLevel()	1011
12.255.2.5 GetRequiredTags()	1011
12.255.2.6 GetUniqueTags()	1011
12.256 gdcm::QueryPatient Class Reference	1012
12.256.1 Detailed Description	1013
12.256.2 Member Function Documentation	1013
12.256.2.1 GetHierarchicalSearchTags()	1013
12.256.2.2 GetName()	1013
12.256.2.3 GetOptionalTags()	1013
12.256.2.4 GetQueryLevel()	1013
12.256.2.5 GetRequiredTags()	1014

12.256.2.6	GetUniqueTags()	1014
12.257	gdcm::QuerySeries Class Reference	1014
12.257.1	Detailed Description	1015
12.257.2	Member Function Documentation	1015
12.257.2.1	GetHierachicalSearchTags()	1015
12.257.2.2	GetName()	1016
12.257.2.3	GetOptionalTags()	1016
12.257.2.4	GetQueryLevel()	1016
12.257.2.5	GetRequiredTags()	1016
12.257.2.6	GetUniqueTags()	1016
12.258	gdcm::QueryStudy Class Reference	1017
12.258.1	Detailed Description	1018
12.258.2	Member Function Documentation	1018
12.258.2.1	GetHierachicalSearchTags()	1018
12.258.2.2	GetName()	1018
12.258.2.3	GetOptionalTags()	1018
12.258.2.4	GetQueryLevel()	1018
12.258.2.5	GetRequiredTags()	1019
12.258.2.6	GetUniqueTags()	1019
12.259	gdcm::RAWCodec Class Reference	1019
12.259.1	Detailed Description	1022
12.259.2	Constructor & Destructor Documentation	1022
12.259.2.1	RAWCodec()	1022
12.259.2.2	~RAWCodec()	1022
12.259.3	Member Function Documentation	1022
12.259.3.1	CanCode()	1022
12.259.3.2	CanDecode()	1022
12.259.3.3	Clone()	1023
12.259.3.4	Code()	1023
12.259.3.5	Decode()	1023
12.259.3.6	DecodeByStreams()	1023
12.259.3.7	DecodeBytes()	1023
12.259.3.8	GetHeaderInfo()	1024
12.260	gdcm::Reader Class Reference	1024
12.260.1	Detailed Description	1026
12.260.2	Constructor & Destructor Documentation	1027
12.260.2.1	Reader()	1027
12.260.2.2	~Reader()	1027
12.260.3	Member Function Documentation	1027
12.260.3.1	CanRead()	1027

12.260.3.2	GetFile() [1/2]	1027
12.260.3.3	GetFile() [2/2]	1028
12.260.3.4	GetStreamCurrentPosition()	1028
12.260.3.5	GetStreamPtr()	1028
12.260.3.6	Read()	1029
12.260.3.7	ReadDataSet()	1029
12.260.3.8	ReadMetaInformation()	1029
12.260.3.9	ReadPreamble()	1029
12.260.3.10	ReadSelectedPrivateTags()	1029
12.260.3.11	ReadSelectedTags()	1030
12.260.3.12	ReadUpToTag()	1030
12.260.3.13	SetFile()	1030
12.260.3.14	SetFileName()	1030
12.260.3.15	SetStream()	1031
12.260.4	Friends And Related Symbol Documentation	1031
12.260.4.1	StreamImageReader	1031
12.260.5	Member Data Documentation	1031
12.260.5.1	F	1031
12.261	gdcm::RealWorldValueMappingContent Struct Reference	1032
12.261.1	Member Data Documentation	1032
12.261.1.1	CodeMeaning	1032
12.261.1.2	CodeValue	1032
12.261.1.3	RealWorldValueIntercept	1033
12.261.1.4	RealWorldValueSlope	1033
12.262	gdcm::Region Class Reference	1033
12.262.1	Detailed Description	1034
12.262.2	Constructor & Destructor Documentation	1034
12.262.2.1	Region()	1034
12.262.2.2	~Region()	1034
12.262.3	Member Function Documentation	1034
12.262.3.1	Area()	1034
12.262.3.2	Clone()	1034
12.262.3.3	ComputeBoundingBox()	1034
12.262.3.4	Empty()	1035
12.262.3.5	IsValid()	1035
12.262.3.6	Print()	1035
12.263	gdcm::Rescaler Class Reference	1035
12.263.1	Detailed Description	1036
12.263.2	Constructor & Destructor Documentation	1037
12.263.2.1	Rescaler()	1037

12.263.2.2	~Rescaler()	1037
12.263.3	Member Function Documentation	1037
12.263.3.1	ComputeInterceptSlopePixelType()	1037
12.263.3.2	ComputePixelTypeFromMinMax()	1038
12.263.3.3	GetIntercept()	1038
12.263.3.4	GetSlope()	1038
12.263.3.5	InverseRescale()	1038
12.263.3.6	InverseRescaleFunctionIntoBestFit()	1038
12.263.3.7	Rescale()	1038
12.263.3.8	RescaleFunctionIntoBestFit()	1039
12.263.3.9	SetIntercept()	1039
12.263.3.10	SetMinMaxForPixelType()	1039
12.263.3.11	SetPixelFormat()	1039
12.263.3.12	SetSlope()	1039
12.263.3.13	SetTargetPixelType()	1040
12.263.3.14	SetUseTargetPixelType()	1040
12.264	gdcm::RLECodec Class Reference	1040
12.264.1	Detailed Description	1043
12.264.2	Constructor & Destructor Documentation	1043
12.264.2.1	RLECodec()	1043
12.264.2.2	~RLECodec()	1043
12.264.3	Member Function Documentation	1044
12.264.3.1	AppendFrameEncode()	1044
12.264.3.2	AppendRowEncode()	1044
12.264.3.3	CanCode()	1044
12.264.3.4	CanDecode()	1044
12.264.3.5	Clone()	1044
12.264.3.6	Code()	1045
12.264.3.7	Decode()	1045
12.264.3.8	DecodeByStreams()	1045
12.264.3.9	DecodeExtent()	1045
12.264.3.10	GetBufferLength()	1045
12.264.3.11	GetHeaderInfo()	1046
12.264.3.12	IsFrameEncoder()	1046
12.264.3.13	IsRowEncoder()	1046
12.264.3.14	SetBufferLength()	1046
12.264.3.15	SetLength()	1046
12.264.3.16	StartEncode()	1046
12.264.3.17	StopEncode()	1046
12.264.4	Friends And Related Symbol Documentation	1047

12.264.4.1 ImageRegionReader	1047
12.265 gdcm::network::RoleSelectionSub Class Reference	1047
12.265.1 Detailed Description	1047
12.265.2 Constructor & Destructor Documentation	1047
12.265.2.1 RoleSelectionSub()	1047
12.265.3 Member Function Documentation	1048
12.265.3.1 Print()	1048
12.265.3.2 Read()	1048
12.265.3.3 SetTuple()	1048
12.265.3.4 Size()	1048
12.265.3.5 Write()	1048
12.266 gdcm::Scanner Class Reference	1049
12.266.1 Detailed Description	1052
12.266.2 Member Typedef Documentation	1052
12.266.2.1 ConstIterator	1052
12.266.2.2 MappingType	1052
12.266.2.3 TagToValue	1052
12.266.2.4 TagToValueValueType	1053
12.266.2.5 ValuesType	1053
12.266.3 Constructor & Destructor Documentation	1053
12.266.3.1 Scanner()	1053
12.266.3.2 ~Scanner()	1053
12.266.4 Member Function Documentation	1053
12.266.4.1 AddPrivateTag()	1053
12.266.4.2 AddSkipTag()	1053
12.266.4.3 AddTag()	1054
12.266.4.4 Begin()	1054
12.266.4.5 ClearSkipTags()	1054
12.266.4.6 ClearTags()	1054
12.266.4.7 End()	1054
12.266.4.8 GetAllFileNamesFromTagToValue()	1054
12.266.4.9 GetFilenameFromTagToValue()	1054
12.266.4.10 GetFileNames()	1055
12.266.4.11 GetKeys()	1055
12.266.4.12 GetMapping()	1055
12.266.4.13 GetMappingFromTagToValue()	1055
12.266.4.14 GetMappings()	1055
12.266.4.15 GetOrderedValues()	1055
12.266.4.16 GetValue()	1056
12.266.4.17 GetValues() [1/2]	1056

12.266.4.18	GetValues() [2/2]	1056
12.266.4.19	IsKey()	1056
12.266.4.20	New()	1057
12.266.4.21	Print()	1057
12.266.4.22	PrintTable()	1057
12.266.4.23	ProcessPublicTag()	1057
12.266.4.24	Scan()	1057
12.266.5	Friends And Related Symbol Documentation	1058
12.266.5.1	operator<<	1058
12.267	gdcmm::Scanner2 Class Reference	1058
12.267.1	Detailed Description	1061
12.267.2	Member Typedef Documentation	1062
12.267.2.1	PrivateConstIterator	1062
12.267.2.2	PrivateMappingType	1062
12.267.2.3	PrivateTagToValue	1062
12.267.2.4	PrivateTagToValueValueType	1062
12.267.2.5	PublicConstIterator	1062
12.267.2.6	PublicMappingType	1062
12.267.2.7	PublicTagToValue	1063
12.267.2.8	PublicTagToValueValueType	1063
12.267.2.9	ValuesType	1063
12.267.3	Constructor & Destructor Documentation	1063
12.267.3.1	Scanner2()	1063
12.267.3.2	~Scanner2()	1063
12.267.4	Member Function Documentation	1063
12.267.4.1	AddPrivateTag()	1063
12.267.4.2	AddPublicTag()	1063
12.267.4.3	AddSkipTag()	1064
12.267.4.4	Begin()	1064
12.267.4.5	ClearPrivateTags()	1064
12.267.4.6	ClearPublicTags()	1064
12.267.4.7	ClearSkipTags()	1064
12.267.4.8	End()	1064
12.267.4.9	GetAllFileNamesFromPrivateTagToValue()	1064
12.267.4.10	GetAllFileNamesFromPublicTagToValue()	1064
12.267.4.11	GetFilenameFromPrivateTagToValue()	1065
12.267.4.12	GetFilenameFromPublicTagToValue()	1065
12.267.4.13	GetFileNames()	1065
12.267.4.14	GetKeys()	1065
12.267.4.15	GetMappingFromPrivateTagToValue()	1065

12.267.4.16	GetMappingFromPublicTagToValue()	1065
12.267.4.17	GetPrivateMapping()	1065
12.267.4.18	GetPrivateMappings()	1066
12.267.4.19	GetPrivateOrderedValues()	1066
12.267.4.20	GetPrivateValue()	1066
12.267.4.21	GetPrivateValues()	1066
12.267.4.22	GetPublicMapping()	1066
12.267.4.23	GetPublicMappings()	1066
12.267.4.24	GetPublicOrderedValues()	1066
12.267.4.25	GetPublicValue()	1067
12.267.4.26	GetPublicValues()	1067
12.267.4.27	GetValues()	1067
12.267.4.28	IsKey()	1067
12.267.4.29	New()	1067
12.267.4.30	Print()	1068
12.267.4.31	PrintTable()	1068
12.267.4.32	PrivateBegin()	1068
12.267.4.33	PrivateEnd()	1068
12.267.4.34	ProcessPrivateTag()	1068
12.267.4.35	ProcessPublicTag()	1068
12.267.4.36	Scan()	1068
12.267.5	Friends And Related Symbol Documentation	1069
12.267.5.1	operator<<	1069
12.268	gdcm::Segment Class Reference	1069
12.268.1	Detailed Description	1071
12.268.2	Member Typedef Documentation	1072
12.268.2.1	BasicCodedEntryVector	1072
12.268.2.2	SurfaceVector	1072
12.268.3	Member Enumeration Documentation	1072
12.268.3.1	ALGOType	1072
12.268.4	Constructor & Destructor Documentation	1072
12.268.4.1	Segment()	1072
12.268.4.2	~Segment()	1072
12.268.5	Member Function Documentation	1072
12.268.5.1	AddSurface()	1072
12.268.5.2	GetALGOType()	1073
12.268.5.3	GetALGOTypeString()	1073
12.268.5.4	GetAnatomicRegion() [1/2]	1073
12.268.5.5	GetAnatomicRegion() [2/2]	1073
12.268.5.6	GetAnatomicRegionModifiers() [1/2]	1073

12.268.5.7	GetAnatomicRegionModifiers() [2/2]	1073
12.268.5.8	GetPropertyCategory() [1/2]	1073
12.268.5.9	GetPropertyCategory() [2/2]	1073
12.268.5.10	GetPropertyType() [1/2]	1073
12.268.5.11	GetPropertyType() [2/2]	1074
12.268.5.12	GetPropertyTypeModifiers() [1/2]	1074
12.268.5.13	GetPropertyTypeModifiers() [2/2]	1074
12.268.5.14	GetSegmentAlgorithmName()	1074
12.268.5.15	GetSegmentAlgorithmType()	1074
12.268.5.16	GetSegmentDescription()	1074
12.268.5.17	GetSegmentLabel()	1074
12.268.5.18	GetSegmentNumber()	1074
12.268.5.19	GetSurface()	1074
12.268.5.20	GetSurfaceCount()	1075
12.268.5.21	GetSurfaces() [1/2]	1075
12.268.5.22	GetSurfaces() [2/2]	1075
12.268.5.23	SetAnatomicRegion()	1075
12.268.5.24	SetAnatomicRegionModifiers()	1075
12.268.5.25	SetPropertyCategory()	1075
12.268.5.26	SetPropertyType()	1075
12.268.5.27	SetPropertyTypeModifiers()	1075
12.268.5.28	SetSegmentAlgorithmName()	1075
12.268.5.29	SetSegmentAlgorithmType() [1/2]	1076
12.268.5.30	SetSegmentAlgorithmType() [2/2]	1076
12.268.5.31	SetSegmentDescription()	1076
12.268.5.32	SetSegmentLabel()	1076
12.268.5.33	SetSegmentNumber()	1076
12.268.5.34	SetSurfaceCount()	1076
12.268.6	Member Data Documentation	1076
12.268.6.1	AnatomicRegion	1076
12.268.6.2	AnatomicRegionModifiers	1076
12.268.6.3	PropertyCategory	1077
12.268.6.4	PropertyType	1077
12.268.6.5	PropertyTypeModifiers	1077
12.268.6.6	SegmentAlgorithmName	1077
12.268.6.7	SegmentAlgorithmType	1077
12.268.6.8	SegmentDescription	1077
12.268.6.9	SegmentLabel	1077
12.268.6.10	SegmentNumber	1077
12.268.6.11	SurfaceCount	1077

12.268.6.12 Surfaces	1078
12.269 gdcmm::SegmentedPaletteColorLookupTable Class Reference	1078
12.269.1 Detailed Description	1081
12.269.2 Constructor & Destructor Documentation	1081
12.269.2.1 SegmentedPaletteColorLookupTable()	1081
12.269.2.2 ~SegmentedPaletteColorLookupTable()	1081
12.269.3 Member Function Documentation	1081
12.269.3.1 Print()	1081
12.269.3.2 SetLUT()	1081
12.270 gdcmm::SegmentReader Class Reference	1082
12.270.1 Detailed Description	1085
12.270.2 Member Typedef Documentation	1085
12.270.2.1 SegmentMap	1085
12.270.2.2 SegmentVector	1085
12.270.3 Constructor & Destructor Documentation	1085
12.270.3.1 SegmentReader()	1085
12.270.3.2 ~SegmentReader()	1085
12.270.4 Member Function Documentation	1085
12.270.4.1 GetSegments() [1/2]	1085
12.270.4.2 GetSegments() [2/2]	1085
12.270.4.3 Read()	1086
12.270.4.4 ReadSegment()	1086
12.270.4.5 ReadSegments()	1086
12.270.5 Member Data Documentation	1086
12.270.5.1 Segments	1086
12.271 gdcmm::SegmentWriter Class Reference	1086
12.271.1 Detailed Description	1089
12.271.2 Member Typedef Documentation	1090
12.271.2.1 SegmentVector	1090
12.271.3 Constructor & Destructor Documentation	1090
12.271.3.1 SegmentWriter()	1090
12.271.3.2 ~SegmentWriter()	1090
12.271.4 Member Function Documentation	1090
12.271.4.1 AddSegment()	1090
12.271.4.2 GetNumberOfSegments()	1090
12.271.4.3 GetSegment()	1090
12.271.4.4 GetSegments() [1/2]	1090
12.271.4.5 GetSegments() [2/2]	1090
12.271.4.6 PrepareWrite()	1091
12.271.4.7 SetNumberOfSegments()	1091

12.271.4.8 SetSegments()	1091
12.271.4.9 Write()	1091
12.271.5 Member Data Documentation	1091
12.271.5.1 Segments	1091
12.272 gdcm::SequenceOfFragments Class Reference	1092
12.272.1 Detailed Description	1094
12.272.2 Member Typedef Documentation	1094
12.272.2.1 ConstIterator	1094
12.272.2.2 FragmentVector	1094
12.272.2.3 Iterator	1094
12.272.2.4 SizeType	1095
12.272.3 Constructor & Destructor Documentation	1095
12.272.3.1 SequenceOfFragments()	1095
12.272.4 Member Function Documentation	1095
12.272.4.1 AddFragment()	1095
12.272.4.2 Begin() [1/2]	1095
12.272.4.3 Begin() [2/2]	1095
12.272.4.4 Clear()	1095
12.272.4.5 ComputeByteLength()	1096
12.272.4.6 ComputeLength()	1096
12.272.4.7 End() [1/2]	1096
12.272.4.8 End() [2/2]	1096
12.272.4.9 GetBuffer()	1096
12.272.4.10 GetFragBuffer()	1096
12.272.4.11 GetFragment()	1096
12.272.4.12 GetLength()	1097
12.272.4.13 GetNumberOfFragments()	1097
12.272.4.14 GetTable() [1/2]	1097
12.272.4.15 GetTable() [2/2]	1097
12.272.4.16 New()	1097
12.272.4.17 operator==()	1097
12.272.4.18 Print()	1098
12.272.4.19 Read()	1098
12.272.4.20 ReadPreValue()	1098
12.272.4.21 ReadValue()	1098
12.272.4.22 SetLength()	1098
12.272.4.23 Write()	1099
12.272.4.24 WriteBuffer()	1099
12.273 gdcm::SequenceOfItems Class Reference	1099
12.273.1 Detailed Description	1102

12.273.2 Member Typedef Documentation	1102
12.273.2.1 ConstIterator	1102
12.273.2.2 ItemVector	1102
12.273.2.3 Iterator	1103
12.273.2.4 SizeType	1103
12.273.3 Constructor & Destructor Documentation	1103
12.273.3.1 SequenceOfItems()	1103
12.273.4 Member Function Documentation	1103
12.273.4.1 AddItem()	1103
12.273.4.2 AddNewUndefinedLengthItem()	1103
12.273.4.3 Begin() [1/2]	1104
12.273.4.4 Begin() [2/2]	1104
12.273.4.5 Clear()	1104
12.273.4.6 ComputeLength()	1104
12.273.4.7 End() [1/2]	1104
12.273.4.8 End() [2/2]	1104
12.273.4.9 FindDataElement()	1104
12.273.4.10 GetItem() [1/2]	1105
12.273.4.11 GetItem() [2/2]	1105
12.273.4.12 GetLength()	1105
12.273.4.13 GetNumberOfItems()	1105
12.273.4.14 IsEmpty()	1105
12.273.4.15 IsUndefinedLength()	1106
12.273.4.16 New()	1106
12.273.4.17 operator=()	1106
12.273.4.18 operator==()	1106
12.273.4.19 Print()	1106
12.273.4.20 Read()	1107
12.273.4.21 RemoveItemByIndex()	1107
12.273.4.22 SetLength()	1107
12.273.4.23 SetLengthToUndefined()	1107
12.273.4.24 SetNumberOfItems()	1107
12.273.4.25 Write()	1107
12.273.5 Member Data Documentation	1108
12.273.5.1 Items	1108
12.273.5.2 SequenceLengthField	1108
12.274 gdcmm::SerieHelper Class Reference	1108
12.274.1 Detailed Description	1110
12.274.2 Member Typedef Documentation	1110
12.274.2.1 Rule	1110

12.274.2.2	SeriesRestrictions	1110
12.274.2.3	SingleSerieUIDFileSetmap	1110
12.274.3	Constructor & Destructor Documentation	1111
12.274.3.1	SerieHelper()	1111
12.274.3.2	~SerieHelper()	1111
12.274.4	Member Function Documentation	1111
12.274.4.1	AddFile()	1111
12.274.4.2	AddFileName()	1111
12.274.4.3	AddRestriction() [1/3]	1111
12.274.4.4	AddRestriction() [2/3]	1111
12.274.4.5	AddRestriction() [3/3]	1111
12.274.4.6	Clear()	1112
12.274.4.7	CreateDefaultUniqueSeriesIdentifier()	1112
12.274.4.8	CreateUniqueSeriesIdentifier()	1112
12.274.4.9	FileNameOrdering()	1112
12.274.4.10	GetFirstSingleSerieUIDFileSet()	1112
12.274.4.11	GetNextSingleSerieUIDFileSet()	1112
12.274.4.12	ImageNumberOrdering()	1112
12.274.4.13	ImagePositionPatientOrdering()	1112
12.274.4.14	OrderFileList()	1112
12.274.4.15	SetDirectory()	1113
12.274.4.16	SetLoadMode()	1113
12.274.4.17	SetUseSeriesDetails()	1113
12.274.4.18	UserOrdering()	1113
12.274.5	Member Data Documentation	1113
12.274.5.1	ItFileSetHt	1113
12.274.5.2	SingleSerieUIDFileSetHT	1113
12.275	gdcm::Series Class Reference	1113
12.275.1	Detailed Description	1114
12.275.2	Constructor & Destructor Documentation	1114
12.275.2.1	Series()	1114
12.276	gdcm::network::ServiceClassApplicationInformation Class Reference	1114
12.276.1	Detailed Description	1114
12.276.2	Constructor & Destructor Documentation	1114
12.276.2.1	ServiceClassApplicationInformation()	1114
12.276.3	Member Function Documentation	1115
12.276.3.1	Print()	1115
12.276.3.2	Read()	1115
12.276.3.3	SetTuple()	1115
12.276.3.4	Size()	1115

12.276.3.5 Write()	1115
12.277 gdcm::ServiceClassUser Class Reference	1116
12.277.1 Detailed Description	1118
12.277.2 Constructor & Destructor Documentation	1119
12.277.2.1 ServiceClassUser() [1/2]	1119
12.277.2.2 ~ServiceClassUser()	1119
12.277.2.3 ServiceClassUser() [2/2]	1119
12.277.3 Member Function Documentation	1119
12.277.3.1 GetAETitle()	1119
12.277.3.2 GetCalledAETitle()	1119
12.277.3.3 GetTimeout()	1119
12.277.3.4 InitializeConnection()	1120
12.277.3.5 IsPresentationContextAccepted()	1120
12.277.3.6 New()	1120
12.277.3.7 operator=()	1120
12.277.3.8 SendEcho()	1120
12.277.3.9 SendFind()	1120
12.277.3.10 SendMove() [1/3]	1121
12.277.3.11 SendMove() [2/3]	1121
12.277.3.12 SendMove() [3/3]	1121
12.277.3.13 SendStore() [1/3]	1121
12.277.3.14 SendStore() [2/3]	1121
12.277.3.15 SendStore() [3/3]	1122
12.277.3.16 SetAETitle()	1122
12.277.3.17 SetCalledAETitle()	1122
12.277.3.18 SetHostname()	1122
12.277.3.19 SetPort()	1122
12.277.3.20 SetPortSCP()	1123
12.277.3.21 SetPresentationContexts()	1123
12.277.3.22 SetTimeout()	1123
12.277.3.23 StartAssociation()	1123
12.277.3.24 StopAssociation()	1124
12.278 gdcm::SHA1 Class Reference	1124
12.278.1 Detailed Description	1124
12.278.2 Constructor & Destructor Documentation	1125
12.278.2.1 SHA1() [1/2]	1125
12.278.2.2 ~SHA1()	1125
12.278.2.3 SHA1() [2/2]	1125
12.278.3 Member Function Documentation	1125
12.278.3.1 Compute()	1125

12.278.3.2	ComputeFile()	1125
12.278.3.3	operator=()	1125
12.279	gdcmm::SimpleMemberCommand< T > Class Template Reference	1126
12.279.1	Detailed Description	1129
12.279.2	Member Typedef Documentation	1129
12.279.2.1	Self	1129
12.279.2.2	TMemberFunctionPointer	1129
12.279.3	Constructor & Destructor Documentation	1129
12.279.3.1	SimpleMemberCommand() [1/2]	1129
12.279.3.2	SimpleMemberCommand() [2/2]	1129
12.279.3.3	~SimpleMemberCommand()	1129
12.279.4	Member Function Documentation	1130
12.279.4.1	Execute() [1/2]	1130
12.279.4.2	Execute() [2/2]	1130
12.279.4.3	New()	1130
12.279.4.4	operator=()	1130
12.279.4.5	SetCallbackFunction()	1130
12.279.5	Member Data Documentation	1131
12.279.5.1	m_MemberFunction	1131
12.279.5.2	m_This	1131
12.280	gdcmm::SimpleSubjectWatcher Class Reference	1131
12.280.1	Detailed Description	1132
12.280.2	Constructor & Destructor Documentation	1132
12.280.2.1	SimpleSubjectWatcher() [1/2]	1132
12.280.2.2	~SimpleSubjectWatcher()	1132
12.280.2.3	SimpleSubjectWatcher() [2/2]	1132
12.280.3	Member Function Documentation	1132
12.280.3.1	EndFilter()	1132
12.280.3.2	operator=()	1132
12.280.3.3	ShowAbort()	1133
12.280.3.4	ShowAnonymization()	1133
12.280.3.5	ShowData()	1133
12.280.3.6	ShowDataSet()	1133
12.280.3.7	ShowFileName()	1133
12.280.3.8	ShowIteration()	1133
12.280.3.9	ShowProgress()	1133
12.280.3.10	StartFilter()	1134
12.280.3.11	TestAbortOff()	1134
12.280.3.12	TestAbortOn()	1134
12.281	gdcmm::MrProtocol::Slice Struct Reference	1134

12.281.1 Member Data Documentation	1135
12.281.1.1 Normal	1135
12.281.1.2 Position	1135
12.282 gdcm::MrProtocol::SliceArray Struct Reference	1135
12.282.1 Member Data Documentation	1136
12.282.1.1 Slices	1136
12.283 gdcm::SmartPointer< ObjectType > Class Template Reference	1136
12.283.1 Detailed Description	1138
12.283.2 Constructor & Destructor Documentation	1138
12.283.2.1 SmartPointer() [1/4]	1138
12.283.2.2 SmartPointer() [2/4]	1138
12.283.2.3 SmartPointer() [3/4]	1139
12.283.2.4 SmartPointer() [4/4]	1139
12.283.2.5 ~SmartPointer()	1139
12.283.3 Member Function Documentation	1139
12.283.3.1 GetPointer()	1139
12.283.3.2 operator ObjectType *()	1139
12.283.3.3 operator*()	1139
12.283.3.4 operator->()	1139
12.283.3.5 operator=() [1/3]	1140
12.283.3.6 operator=() [2/3]	1140
12.283.3.7 operator=() [3/3]	1140
12.284 gdcm::network::SOPClassExtendedNegotiationSub Class Reference	1140
12.284.1 Detailed Description	1141
12.284.2 Constructor & Destructor Documentation	1141
12.284.2.1 SOPClassExtendedNegotiationSub()	1141
12.284.3 Member Function Documentation	1141
12.284.3.1 Print()	1141
12.284.3.2 Read()	1141
12.284.3.3 SetTuple()	1141
12.284.3.4 Size()	1141
12.284.3.5 Write()	1141
12.285 gdcm::SOPClassUIDToIOD Class Reference	1142
12.285.1 Detailed Description	1142
12.285.2 Member Typedef Documentation	1142
12.285.2.1 const	1142
12.285.3 Member Function Documentation	1142
12.285.3.1 GetIOD()	1142
12.285.3.2 GetIODFromSOPClassUID()	1143
12.285.3.3 GetNumberOfSOPClassToIOD()	1143

12.285.3.4	GetSOPClassUIDFromIOD()	1143
12.285.3.5	GetSOPClassUIDToIOD()	1143
12.285.3.6	GetSOPClassUIDToIODs()	1143
12.286	gdcm::Sorter Class Reference	1143
12.286.1	Detailed Description	1145
12.286.2	Member Typedef Documentation	1145
12.286.2.1	SelectionMap	1145
12.286.2.2	SortFunction	1145
12.286.3	Constructor & Destructor Documentation	1145
12.286.3.1	Sorter()	1145
12.286.3.2	~Sorter()	1146
12.286.4	Member Function Documentation	1146
12.286.4.1	AddSelect()	1146
12.286.4.2	GetFileNames()	1146
12.286.4.3	Print()	1146
12.286.4.4	SetSortFunction()	1146
12.286.4.5	SetTagsToRead()	1147
12.286.4.6	Sort()	1147
12.286.4.7	StableSort()	1147
12.286.5	Friends And Related Symbol Documentation	1147
12.286.5.1	operator<<	1147
12.286.6	Member Data Documentation	1147
12.286.6.1	FileNames	1147
12.286.6.2	Selection	1148
12.286.6.3	SortFunc	1148
12.286.6.4	TagsToRead	1148
12.287	gdcm::Spacing Class Reference	1148
12.287.1	Detailed Description	1149
12.287.2	Member Enumeration Documentation	1150
12.287.2.1	SpacingType	1150
12.287.3	Constructor & Destructor Documentation	1150
12.287.3.1	Spacing()	1150
12.287.3.2	~Spacing()	1150
12.287.4	Member Function Documentation	1150
12.287.4.1	ComputePixelAspectRatioFromPixelSpacing()	1150
12.288	gdcm::Spectroscopy Class Reference	1150
12.288.1	Detailed Description	1151
12.288.2	Constructor & Destructor Documentation	1151
12.288.2.1	Spectroscopy()	1151
12.289	gdcm::SplitMosaicFilter Class Reference	1151

12.289.1 Detailed Description	1152
12.289.2 Constructor & Destructor Documentation	1152
12.289.2.1 SplitMosaicFilter()	1152
12.289.2.2 ~SplitMosaicFilter()	1152
12.289.3 Member Function Documentation	1152
12.289.3.1 ComputeCSAImageHeaderInfo()	1152
12.289.3.2 ComputeCSASeriesHeaderInfo()	1153
12.289.3.3 ComputeMOSAICDimensions()	1153
12.289.3.4 ComputeMOSAICImagePositionPatient()	1153
12.289.3.5 ComputeMOSAICSliceNormal()	1153
12.289.3.6 ComputeMOSAICSlicePosition()	1153
12.289.3.7 GetAcquisitionSize()	1154
12.289.3.8 GetFile() [1/2]	1154
12.289.3.9 GetFile() [2/2]	1154
12.289.3.10 GetImage() [1/2]	1154
12.289.3.11 GetImage() [2/2]	1154
12.289.3.12 GetNumberOfImagesInMosaic()	1154
12.289.3.13 SetFile()	1154
12.289.3.14 SetImage()	1154
12.289.3.15 Split()	1155
12.290 gdcm::StartEvent Class Reference	1155
12.291 gdcm::static_assert_test< x > Struct Template Reference	1156
12.292 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference	1157
12.293 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference	1157
12.293.1 Member Enumeration Documentation	1158
12.293.1.1 anonymous enum	1158
12.294 gdcm::StreamImageReader Class Reference	1158
12.294.1 Detailed Description	1159
12.294.2 Constructor & Destructor Documentation	1159
12.294.2.1 StreamImageReader()	1159
12.294.2.2 ~StreamImageReader()	1159
12.294.3 Member Function Documentation	1160
12.294.3.1 CanReadImage()	1160
12.294.3.2 DefinePixelExtent()	1160
12.294.3.3 DefineProperBufferLength()	1160
12.294.3.4 GetDimensionsValueForResolution()	1161
12.294.3.5 GetFile()	1161
12.294.3.6 Read()	1161
12.294.3.7 ReadImageInformation()	1161
12.294.3.8 SetFileName()	1162

12.294.3.9 SetStream()	1162
12.295 gdcm::StreamImageWriter Class Reference	1162
12.295.1 Detailed Description	1164
12.295.2 Constructor & Destructor Documentation	1164
12.295.2.1 StreamImageWriter()	1164
12.295.2.2 ~StreamImageWriter()	1164
12.295.3 Member Function Documentation	1165
12.295.3.1 CanWriteFile()	1165
12.295.3.2 DefinePixelExtent()	1165
12.295.3.3 DefineProperBufferLength()	1165
12.295.3.4 SetFile()	1166
12.295.3.5 SetFileName()	1166
12.295.3.6 SetStream()	1166
12.295.3.7 Write()	1166
12.295.3.8 WriteImageInformation()	1167
12.295.3.9 WriteImageSubregionRAW()	1167
12.295.3.10 WriteRawHeader()	1167
12.295.4 Member Data Documentation	1167
12.295.4.1 mElementOffsets	1167
12.295.4.2 mElementOffsets1	1167
12.295.4.3 mspFile	1168
12.295.4.4 mWriter	1168
12.295.4.5 mXMax	1168
12.295.4.6 mXMin	1168
12.295.4.7 mYMax	1168
12.295.4.8 mYMin	1168
12.295.4.9 mZMax	1168
12.295.4.10 mZMin	1168
12.296 gdcm::StrictScanner Class Reference	1169
12.296.1 Detailed Description	1172
12.296.2 Member Typedef Documentation	1172
12.296.2.1 ConstIterator	1172
12.296.2.2 MappingType	1172
12.296.2.3 TagToValue	1173
12.296.2.4 TagToValueValueType	1173
12.296.2.5 ValuesType	1173
12.296.3 Constructor & Destructor Documentation	1173
12.296.3.1 StrictScanner()	1173
12.296.3.2 ~StrictScanner()	1173
12.296.4 Member Function Documentation	1173

12.296.4.1	AddPrivateTag()	1173
12.296.4.2	AddSkipTag()	1174
12.296.4.3	AddTag()	1174
12.296.4.4	Begin()	1174
12.296.4.5	ClearSkipTags()	1174
12.296.4.6	ClearTags()	1174
12.296.4.7	End()	1174
12.296.4.8	GetAllFileNamesFromTagToValue()	1174
12.296.4.9	GetFilenameFromTagToValue()	1175
12.296.4.10	GetFileNames()	1175
12.296.4.11	GetKeys()	1175
12.296.4.12	GetMapping()	1175
12.296.4.13	GetMappingFromTagToValue()	1175
12.296.4.14	GetMappings()	1175
12.296.4.15	GetOrderedValues()	1176
12.296.4.16	GetValue()	1176
12.296.4.17	GetValues() [1/2]	1176
12.296.4.18	GetValues() [2/2]	1176
12.296.4.19	IsKey()	1176
12.296.4.20	New()	1177
12.296.4.21	Print()	1177
12.296.4.22	PrintTable()	1177
12.296.4.23	ProcessPublicTag()	1177
12.296.4.24	Scan()	1177
12.296.5	Friends And Related Symbol Documentation	1178
12.296.5.1	operator<<	1178
12.297	gdcmm::StrictScanner2 Class Reference	1178
12.297.1	Detailed Description	1181
12.297.2	Member Typedef Documentation	1182
12.297.2.1	PrivateConstIterator	1182
12.297.2.2	PrivateMappingType	1182
12.297.2.3	PrivateTagToValue	1182
12.297.2.4	PrivateTagToValueValueType	1182
12.297.2.5	PublicConstIterator	1182
12.297.2.6	PublicMappingType	1182
12.297.2.7	PublicTagToValue	1182
12.297.2.8	PublicTagToValueValueType	1182
12.297.2.9	ValuesType	1182
12.297.3	Constructor & Destructor Documentation	1183
12.297.3.1	StrictScanner2()	1183

12.297.3.2 ~StrictScanner2()	1183
12.297.4 Member Function Documentation	1183
12.297.4.1 AddPrivateTag()	1183
12.297.4.2 AddPublicTag()	1183
12.297.4.3 AddSkipTag()	1183
12.297.4.4 Begin()	1183
12.297.4.5 ClearPrivateTags()	1183
12.297.4.6 ClearPublicTags()	1184
12.297.4.7 ClearSkipTags()	1184
12.297.4.8 End()	1184
12.297.4.9 GetAllFileNamesFromPrivateTagToValue()	1184
12.297.4.10 GetAllFileNamesFromPublicTagToValue()	1184
12.297.4.11 GetFilenameFromPrivateTagToValue()	1184
12.297.4.12 GetFilenameFromPublicTagToValue()	1184
12.297.4.13 GetFileNames()	1185
12.297.4.14 GetKeys()	1185
12.297.4.15 GetMappingFromPrivateTagToValue()	1185
12.297.4.16 GetMappingFromPublicTagToValue()	1185
12.297.4.17 GetPrivateMapping()	1185
12.297.4.18 GetPrivateMappings()	1185
12.297.4.19 GetPrivateOrderedValues()	1185
12.297.4.20 GetPrivateValue()	1186
12.297.4.21 GetPrivateValues()	1186
12.297.4.22 GetPublicMapping()	1186
12.297.4.23 GetPublicMappings()	1186
12.297.4.24 GetPublicOrderedValues()	1186
12.297.4.25 GetPublicValue()	1186
12.297.4.26 GetPublicValues()	1187
12.297.4.27 GetValues()	1187
12.297.4.28 IsKey()	1187
12.297.4.29 New()	1187
12.297.4.30 Print()	1187
12.297.4.31 PrintTable()	1187
12.297.4.32 PrivateBegin()	1188
12.297.4.33 PrivateEnd()	1188
12.297.4.34 ProcessPrivateTag()	1188
12.297.4.35 ProcessPublicTag()	1188
12.297.4.36 Scan()	1188
12.297.5 Friends And Related Symbol Documentation	1188
12.297.5.1 operator<<	1188

12.298 gdcmm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	1189
12.298.1 Detailed Description	1190
12.298.2 Member Typedef Documentation	1190
12.298.2.1 const_iterator	1190
12.298.2.2 const_reference	1191
12.298.2.3 const_reverse_iterator	1191
12.298.2.4 difference_type	1191
12.298.2.5 iterator	1191
12.298.2.6 pointer	1191
12.298.2.7 reference	1191
12.298.2.8 reverse_iterator	1191
12.298.2.9 size_type	1191
12.298.2.10 value_type	1192
12.298.3 Constructor & Destructor Documentation	1192
12.298.3.1 String() [1/4]	1192
12.298.3.2 String() [2/4]	1192
12.298.3.3 String() [3/4]	1192
12.298.3.4 String() [4/4]	1192
12.298.4 Member Function Documentation	1192
12.298.4.1 IsValid()	1192
12.298.4.2 operator const char *()	1193
12.298.4.3 Trim() [1/2]	1193
12.298.4.4 Trim() [2/2]	1193
12.298.4.5 Truncate()	1193
12.299 gdcmm::StringFilter Class Reference	1193
12.299.1 Detailed Description	1194
12.299.2 Constructor & Destructor Documentation	1194
12.299.2.1 StringFilter()	1194
12.299.2.2 ~StringFilter()	1195
12.299.3 Member Function Documentation	1195
12.299.3.1 ExecuteQuery() [1/2]	1195
12.299.3.2 ExecuteQuery() [2/2]	1195
12.299.3.3 FromString()	1195
12.299.3.4 GetFile() [1/2]	1195
12.299.3.5 GetFile() [2/2]	1195
12.299.3.6 SetDicts()	1195
12.299.3.7 SetFile()	1196
12.299.3.8 ToString() [1/3]	1196
12.299.3.9 ToString() [2/3]	1196
12.299.3.10 ToString() [3/3]	1196

12.299.3.11 ToStringPair() [1/3]	1196
12.299.3.12 ToStringPair() [2/3]	1197
12.299.3.13 ToStringPair() [3/3]	1197
12.299.3.14 UseDictAlways()	1197
12.300 gdcmm::Study Class Reference	1197
12.300.1 Detailed Description	1197
12.300.2 Constructor & Destructor Documentation	1197
12.300.2.1 Study()	1197
12.301 gdcmm::Subject Class Reference	1198
12.301.1 Detailed Description	1199
12.301.2 Constructor & Destructor Documentation	1199
12.301.2.1 Subject()	1199
12.301.2.2 ~Subject()	1200
12.301.3 Member Function Documentation	1200
12.301.3.1 AddObserver() [1/2]	1200
12.301.3.2 AddObserver() [2/2]	1200
12.301.3.3 GetCommand()	1200
12.301.3.4 HasObserver()	1200
12.301.3.5 InvokeEvent() [1/2]	1200
12.301.3.6 InvokeEvent() [2/2]	1201
12.301.3.7 RemoveAllObservers()	1201
12.301.3.8 RemoveObserver()	1201
12.302 gdcmm::Surface Class Reference	1201
12.302.1 Detailed Description	1204
12.302.2 Member Enumeration Documentation	1204
12.302.2.1 STATES	1204
12.302.2.2 VIEWType	1205
12.302.3 Constructor & Destructor Documentation	1205
12.302.3.1 Surface()	1205
12.302.3.2 ~Surface()	1205
12.302.4 Member Function Documentation	1205
12.302.4.1 GetAlgorithmFamily() [1/2]	1205
12.302.4.2 GetAlgorithmFamily() [2/2]	1205
12.302.4.3 GetAlgorithmName()	1206
12.302.4.4 GetAlgorithmVersion()	1206
12.302.4.5 GetAxisOfRotation()	1206
12.302.4.6 GetCenterOfRotation()	1206
12.302.4.7 GetFiniteVolume()	1206
12.302.4.8 GetManifold()	1206
12.302.4.9 GetMaximumPointDistance()	1206

12.302.4.10	GetMeanPointDistance()	1206
12.302.4.11	GetMeshPrimitive() [1/2]	1207
12.302.4.12	GetMeshPrimitive() [2/2]	1207
12.302.4.13	GetNumberOfSurfacePoints()	1207
12.302.4.14	GetNumberOfVectors()	1207
12.302.4.15	GetPointCoordinatesData() [1/2]	1207
12.302.4.16	GetPointCoordinatesData() [2/2]	1207
12.302.4.17	GetPointPositionAccuracy()	1207
12.302.4.18	GetPointsBoundingBoxCoordinates()	1207
12.302.4.19	GetProcessingAlgorithm() [1/2]	1208
12.302.4.20	GetProcessingAlgorithm() [2/2]	1208
12.302.4.21	GetRecommendedDisplayCIELabValue() [1/2]	1208
12.302.4.22	GetRecommendedDisplayCIELabValue() [2/2]	1208
12.302.4.23	GetRecommendedDisplayGrayscaleValue()	1208
12.302.4.24	GetRecommendedPresentationOpacity()	1208
12.302.4.25	GetRecommendedPresentationType()	1208
12.302.4.26	GetSTATES()	1208
12.302.4.27	GetSTATESString()	1208
12.302.4.28	GetSurfaceComments()	1209
12.302.4.29	GetSurfaceNumber()	1209
12.302.4.30	GetSurfaceProcessing()	1209
12.302.4.31	GetSurfaceProcessingDescription()	1209
12.302.4.32	GetSurfaceProcessingRatio()	1209
12.302.4.33	GetVectorAccuracy()	1209
12.302.4.34	GetVectorCoordinateData() [1/2]	1209
12.302.4.35	GetVectorCoordinateData() [2/2]	1209
12.302.4.36	GetVectorDimensionality()	1209
12.302.4.37	GetVIEWType()	1209
12.302.4.38	GetVIEWTypeString()	1210
12.302.4.39	SetAlgorithmFamily()	1210
12.302.4.40	SetAlgorithmName()	1210
12.302.4.41	SetAlgorithmVersion()	1210
12.302.4.42	SetAxisOfRotation()	1210
12.302.4.43	SetCenterOfRotation()	1210
12.302.4.44	SetFiniteVolume()	1210
12.302.4.45	SetManifold()	1210
12.302.4.46	SetMaximumPointDistance()	1211
12.302.4.47	SetMeanPointDistance()	1211
12.302.4.48	SetMeshPrimitive()	1211
12.302.4.49	SetNumberOfSurfacePoints()	1211

12.302.4.50	SetNumberOfVectors()	1211
12.302.4.51	SetPointCoordinatesData()	1211
12.302.4.52	SetPointPositionAccuracy()	1211
12.302.4.53	SetPointsBoundingBoxCoordinates()	1211
12.302.4.54	SetProcessingAlgorithm()	1212
12.302.4.55	SetRecommendedDisplayCIELabValue() [1/3]	1212
12.302.4.56	SetRecommendedDisplayCIELabValue() [2/3]	1212
12.302.4.57	SetRecommendedDisplayCIELabValue() [3/3]	1212
12.302.4.58	SetRecommendedDisplayGrayscaleValue()	1212
12.302.4.59	SetRecommendedPresentationOpacity()	1212
12.302.4.60	SetRecommendedPresentationType()	1212
12.302.4.61	SetSurfaceComments()	1212
12.302.4.62	SetSurfaceNumber()	1213
12.302.4.63	SetSurfaceProcessing()	1213
12.302.4.64	SetSurfaceProcessingDescription()	1213
12.302.4.65	SetSurfaceProcessingRatio()	1213
12.302.4.66	SetVectorAccuracy()	1213
12.302.4.67	SetVectorCoordinateData()	1213
12.302.4.68	SetVectorDimensionality()	1213
12.303	gdcm::SurfaceHelper Class Reference	1214
12.303.1	Detailed Description	1214
12.303.2	Member Typedef Documentation	1214
12.303.2.1	ColorArray	1214
12.303.3	Member Function Documentation	1215
12.303.3.1	RecommendedDisplayCIELabToRGB() [1/2]	1215
12.303.3.2	RecommendedDisplayCIELabToRGB() [2/2]	1215
12.303.3.3	RGBToRecommendedDisplayCIELab()	1216
12.303.3.4	RGBToRecommendedDisplayGrayscale()	1216
12.304	gdcm::SurfaceReader Class Reference	1217
12.304.1	Detailed Description	1220
12.304.2	Constructor & Destructor Documentation	1220
12.304.2.1	SurfaceReader()	1220
12.304.2.2	~SurfaceReader()	1220
12.304.3	Member Function Documentation	1220
12.304.3.1	GetNumberOfSurfaces()	1220
12.304.3.2	Read()	1221
12.304.3.3	ReadPointMacro()	1221
12.304.3.4	ReadSurface()	1221
12.304.3.5	ReadSurfaces()	1221
12.305	gdcm::SurfaceWriter Class Reference	1221

12.305.1 Detailed Description	1224
12.305.2 Constructor & Destructor Documentation	1224
12.305.2.1 SurfaceWriter()	1224
12.305.2.2 ~SurfaceWriter()	1224
12.305.3 Member Function Documentation	1224
12.305.3.1 ComputeNumberOfSurfaces()	1224
12.305.3.2 GetNumberOfSurfaces()	1225
12.305.3.3 PrepareWrite()	1225
12.305.3.4 PrepareWritePointMacro()	1225
12.305.3.5 SetNumberOfSurfaces()	1225
12.305.3.6 Write()	1225
12.305.4 Member Data Documentation	1225
12.305.4.1 NumberOfSurfaces	1225
12.306 gdcmm::SwapCode Class Reference	1225
12.306.1 Detailed Description	1226
12.306.2 Member Enumeration Documentation	1226
12.306.2.1 SwapCodeType	1226
12.306.3 Constructor & Destructor Documentation	1227
12.306.3.1 SwapCode()	1227
12.306.4 Member Function Documentation	1227
12.306.4.1 GetIndex()	1227
12.306.4.2 GetSwapCodeString()	1227
12.306.4.3 operator SwapCode::SwapCodeType()	1227
12.306.5 Friends And Related Symbol Documentation	1228
12.306.5.1 operator<<	1228
12.307 gdcmm::SwapperDoOp Class Reference	1228
12.307.1 Member Function Documentation	1228
12.307.1.1 Swap()	1228
12.307.1.2 SwapArray()	1228
12.308 gdcmm::SwapperNoOp Class Reference	1229
12.308.1 Detailed Description	1229
12.308.2 Member Function Documentation	1229
12.308.2.1 Swap()	1229
12.308.2.2 SwapArray()	1229
12.309 gdcmm::System Class Reference	1229
12.309.1 Detailed Description	1231
12.309.2 Member Function Documentation	1231
12.309.2.1 ConvertToUNC()	1231
12.309.2.2 DeleteDirectory()	1231
12.309.2.3 EncodeBytes()	1231

12.309.2.4	FileExists()	1231
12.309.2.5	FileIsDirectory()	1232
12.309.2.6	FileIsSymlink()	1232
12.309.2.7	FileSize()	1232
12.309.2.8	FileTime()	1232
12.309.2.9	FormatDateTime()	1233
12.309.2.10	GetCurrentDateTime()	1233
12.309.2.11	GetCurrentModuleFileName()	1233
12.309.2.12	GetCurrentProcessFileName()	1233
12.309.2.13	GetCurrentResourcesDirectory()	1233
12.309.2.14	GetCWD()	1234
12.309.2.15	GetHostName()	1234
12.309.2.16	GetLastError()	1234
12.309.2.17	GetLocaleCharset()	1234
12.309.2.18	GetPermissions()	1234
12.309.2.19	GetTimezoneOffsetFromUTC()	1234
12.309.2.20	MakeDirectory()	1235
12.309.2.21	ParseDateTime() [1/2]	1235
12.309.2.22	ParseDateTime() [2/2]	1235
12.309.2.23	RemoveFile()	1235
12.309.2.24	SetPermissions()	1235
12.309.2.25	StrCaseCmp()	1235
12.309.2.26	StrNCaseCmp()	1236
12.309.2.27	StrSep()	1236
12.309.2.28	StrTokR()	1236
12.310	gdcmm::Table Class Reference	1236
12.310.1	Detailed Description	1238
12.310.2	Member Typedef Documentation	1238
12.310.2.1	MapTableEntry	1238
12.310.3	Constructor & Destructor Documentation	1238
12.310.3.1	Table() [1/2]	1238
12.310.3.2	~Table()	1238
12.310.3.3	Table() [2/2]	1238
12.310.4	Member Function Documentation	1238
12.310.4.1	GetTableEntry()	1238
12.310.4.2	InsertEntry()	1239
12.310.4.3	operator=()	1239
12.310.5	Friends And Related Symbol Documentation	1239
12.310.5.1	operator<<	1239
12.310.6	Member Data Documentation	1239

12.310.6.1 TableInternal	1239
12.311 gdcm::TableEntry Class Reference	1239
12.311.1 Detailed Description	1240
12.311.2 Constructor & Destructor Documentation	1240
12.311.2.1 TableEntry()	1240
12.311.2.2 ~TableEntry()	1240
12.312 gdcm::TableReader Class Reference	1240
12.312.1 Detailed Description	1241
12.312.2 Constructor & Destructor Documentation	1241
12.312.2.1 TableReader()	1241
12.312.2.2 ~TableReader()	1241
12.312.3 Member Function Documentation	1241
12.312.3.1 CharacterDataHandler()	1241
12.312.3.2 EndElement()	1242
12.312.3.3 GetDefs()	1242
12.312.3.4 GetFilename()	1242
12.312.3.5 HandleIOD()	1242
12.312.3.6 HandleIODEntry()	1242
12.312.3.7 HandleMacro()	1242
12.312.3.8 HandleMacroEntry()	1242
12.312.3.9 HandleMacroEntryDescription()	1242
12.312.3.10 HandleModule()	1243
12.312.3.11 HandleModuleEntry()	1243
12.312.3.12 HandleModuleEntryDescription()	1243
12.312.3.13 HandleModuleInclude()	1243
12.312.3.14 Read()	1243
12.312.3.15 SetFilename()	1243
12.312.3.16 StartElement()	1243
12.313 gdcm::network::TableRow Class Reference	1244
12.313.1 Constructor & Destructor Documentation	1244
12.313.1.1 TableRow()	1244
12.313.1.2 ~TableRow()	1245
12.313.2 Member Data Documentation	1245
12.313.2.1 transitions	1245
12.314 gdcm::Tag Class Reference	1245
12.314.1 Detailed Description	1247
12.314.2 Constructor & Destructor Documentation	1248
12.314.2.1 Tag() [1/3]	1248
12.314.2.2 Tag() [2/3]	1248
12.314.2.3 Tag() [3/3]	1248

12.314.3 Member Function Documentation	1248
12.314.3.1 GetElement()	1248
12.314.3.2 GetElementTag()	1249
12.314.3.3 GetGroup()	1249
12.314.3.4 GetLength()	1249
12.314.3.5 GetPrivateCreator()	1249
12.314.3.6 IsGroupLength()	1249
12.314.3.7 IsGroupXX()	1250
12.314.3.8 IsIllegal()	1250
12.314.3.9 IsPrivate()	1250
12.314.3.10 IsPrivateCreator()	1250
12.314.3.11 IsPublic()	1251
12.314.3.12 operator"!="()	1251
12.314.3.13 operator<()	1251
12.314.3.14 operator<=()	1251
12.314.3.15 operator=()	1251
12.314.3.16 operator==()	1251
12.314.3.17 operator[]() [1/2]	1252
12.314.3.18 operator[]() [2/2]	1252
12.314.3.19 PrintAsContinuousString()	1252
12.314.3.20 PrintAsContinuousUpperCaseString()	1252
12.314.3.21 PrintAsPipeSeparatedString()	1252
12.314.3.22 Read()	1252
12.314.3.23 ReadFromCommaSeparatedString()	1253
12.314.3.24 ReadFromContinuousString()	1253
12.314.3.25 ReadFromPipeSeparatedString()	1253
12.314.3.26 SetElement()	1253
12.314.3.27 SetElementTag() [1/2]	1254
12.314.3.28 SetElementTag() [2/2]	1254
12.314.3.29 SetGroup()	1254
12.314.3.30 SetPrivateCreator()	1254
12.314.3.31 Write()	1254
12.314.4 Friends And Related Symbol Documentation	1255
12.314.4.1 operator<<	1255
12.314.4.2 operator>>	1255
12.314.5 Member Data Documentation	1255
12.314.5.1 bytes	1255
12.314.5.2 tag	1255
12.314.5.3 tags	1255
12.315 gdcmm::TagPath Class Reference	1255

12.315.1 Detailed Description	1256
12.315.2 Constructor & Destructor Documentation	1256
12.315.2.1 TagPath()	1256
12.315.2.2 ~TagPath()	1256
12.315.3 Member Function Documentation	1256
12.315.3.1 ConstructFromString()	1256
12.315.3.2 ConstructFromTagList()	1257
12.315.3.3 IsValid()	1257
12.315.3.4 Print()	1257
12.315.3.5 Push() [1/2]	1257
12.315.3.6 Push() [2/2]	1257
12.316 gdcmm::Testing Class Reference	1257
12.316.1 Detailed Description	1258
12.316.2 Member Typedef Documentation	1259
12.316.2.1 MD5DataImagesType	1259
12.316.2.2 MediaStorageDataFileType	1259
12.316.3 Constructor & Destructor Documentation	1259
12.316.3.1 Testing()	1259
12.316.3.2 ~Testing()	1259
12.316.4 Member Function Documentation	1259
12.316.4.1 ComputeFileMD5()	1259
12.316.4.2 ComputeMD5()	1259
12.316.4.3 GetDataExtraRoot()	1260
12.316.4.4 GetDataRoot()	1260
12.316.4.5 GetFileName()	1260
12.316.4.6 GetFileNames()	1260
12.316.4.7 GetLossyFlagFromFile()	1260
12.316.4.8 GetMD5DataImage()	1261
12.316.4.9 GetMD5DataImages()	1261
12.316.4.10 GetMD5FromBrokenFile()	1261
12.316.4.11 GetMD5FromFile()	1261
12.316.4.12 GetMediaStorageDataFile()	1261
12.316.4.13 GetMediaStorageDataFiles()	1261
12.316.4.14 GetMediaStorageFromFile()	1261
12.316.4.15 GetNumberOfFileNames()	1262
12.316.4.16 GetNumberOfMD5DataImages()	1262
12.316.4.17 GetNumberOfMediaStorageDataFiles()	1262
12.316.4.18 GetPixelSpacingDataRoot()	1262
12.316.4.19 GetSelectedPrivateGroupOffsetFromFile()	1262
12.316.4.20 GetSelectedTagsOffsetFromFile()	1262

12.316.4.21	GetSourceDirectory()	1262
12.316.4.22	GetStreamOffsetFromFile()	1263
12.316.4.23	GetTempDirectory()	1263
12.316.4.24	GetTempDirectoryW()	1263
12.316.4.25	GetTempFilename()	1263
12.316.4.26	GetTempFilenameW()	1263
12.316.4.27	Print()	1264
12.317	gdcm::Trace Class Reference	1264
12.317.1	Detailed Description	1265
12.317.2	Constructor & Destructor Documentation	1265
12.317.2.1	Trace()	1265
12.317.2.2	~Trace()	1265
12.317.3	Member Function Documentation	1265
12.317.3.1	DebugOff()	1265
12.317.3.2	DebugOn()	1266
12.317.3.3	ErrorOff()	1266
12.317.3.4	ErrorOn()	1266
12.317.3.5	GetDebugFlag()	1266
12.317.3.6	GetDebugStream()	1266
12.317.3.7	GetErrorFlag()	1266
12.317.3.8	GetErrorStream()	1266
12.317.3.9	GetStream()	1266
12.317.3.10	GetWarningFlag()	1267
12.317.3.11	GetWarningStream()	1267
12.317.3.12	SetDebug()	1267
12.317.3.13	SetDebugStream()	1267
12.317.3.14	SetError()	1267
12.317.3.15	SetErrorStream()	1267
12.317.3.16	SetStream()	1268
12.317.3.17	SetStreamToFile()	1268
12.317.3.18	SetWarning()	1268
12.317.3.19	SetWarningStream()	1268
12.317.3.20	WarningOff()	1268
12.317.3.21	WarningOn()	1269
12.318	gdcm::TransferSyntax Class Reference	1269
12.318.1	Detailed Description	1270
12.318.2	Member Enumeration Documentation	1271
12.318.2.1	NegotiatedType	1271
12.318.2.2	TSType	1271
12.318.3	Constructor & Destructor Documentation	1272

12.318.3.1	TransferSyntax()	1272
12.318.4	Member Function Documentation	1273
12.318.4.1	CanStoreLossy()	1273
12.318.4.2	GetNegociatedType()	1273
12.318.4.3	GetString()	1273
12.318.4.4	GetSwapCode()	1273
12.318.4.5	GetTSSString()	1273
12.318.4.6	GetTSType()	1273
12.318.4.7	IsEncapsulated()	1274
12.318.4.8	IsEncoded()	1274
12.318.4.9	IsExplicit()	1274
12.318.4.10	IsImplicit()	1274
12.318.4.11	IsLossless()	1274
12.318.4.12	IsLossy()	1274
12.318.4.13	IsValid()	1274
12.318.4.14	operator TSType()	1274
12.318.5	Friends And Related Symbol Documentation	1275
12.318.5.1	operator<<	1275
12.319	gdcm::network::TransferSyntaxSub Class Reference	1275
12.319.1	Detailed Description	1275
12.319.2	Constructor & Destructor Documentation	1276
12.319.2.1	TransferSyntaxSub()	1276
12.319.3	Member Function Documentation	1276
12.319.3.1	GetName()	1276
12.319.3.2	operator==()	1276
12.319.3.3	Print()	1276
12.319.3.4	Read()	1276
12.319.3.5	SetName()	1276
12.319.3.6	SetNameFromUID()	1276
12.319.3.7	Size()	1277
12.319.3.8	Write()	1277
12.320	gdcm::network::Transition Struct Reference	1277
12.320.1	Constructor & Destructor Documentation	1278
12.320.1.1	Transition() [1/2]	1278
12.320.1.2	~Transition()	1278
12.320.1.3	Transition() [2/2]	1278
12.320.2	Member Function Documentation	1278
12.320.2.1	MakeNew()	1278
12.320.3	Member Data Documentation	1278
12.320.3.1	mAction	1278

12.320.3.2 mEnd	1279
12.321 gdcmm::Type Class Reference	1279
12.321.1 Detailed Description	1280
12.321.2 Member Enumeration Documentation	1280
12.321.2.1 TypeType	1280
12.321.3 Constructor & Destructor Documentation	1280
12.321.3.1 Type()	1280
12.321.4 Member Function Documentation	1281
12.321.4.1 GetTypeString()	1281
12.321.4.2 GetTypeType()	1281
12.321.4.3 operator TypeType()	1281
12.321.5 Friends And Related Symbol Documentation	1281
12.321.5.1 operator<<	1281
12.322 gdcmm::UI Struct Reference	1281
12.322.1 Friends And Related Symbol Documentation	1282
12.322.1.1 operator<<	1282
12.322.2 Member Data Documentation	1282
12.322.2.1 Internal	1282
12.323 gdcmm::UIDGenerator Class Reference	1282
12.323.1 Detailed Description	1283
12.323.2 Constructor & Destructor Documentation	1283
12.323.2.1 UIDGenerator()	1283
12.323.3 Member Function Documentation	1283
12.323.3.1 Generate()	1283
12.323.3.2 GenerateUUID()	1283
12.323.3.3 GetGDCMUID()	1284
12.323.3.4 GetRoot()	1284
12.323.3.5 IsValid()	1284
12.323.3.6 SetRoot()	1284
12.324 gdcmm::UIDs Class Reference	1285
12.324.1 Detailed Description	1300
12.324.2 Member Typedef Documentation	1301
12.324.2.1 TransferSyntaxStringsType	1301
12.324.3 Member Enumeration Documentation	1301
12.324.3.1 TSName	1301
12.324.3.2 TSType	1311
12.324.4 Constructor & Destructor Documentation	1327
12.324.4.1 UIDs()	1327
12.324.5 Member Function Documentation	1327
12.324.5.1 GetName()	1327

12.324.5.2	GetNumberOfTransferSyntaxStrings()	1328
12.324.5.3	GetString()	1328
12.324.5.4	GetTransferSyntaxString()	1328
12.324.5.5	GetTransferSyntaxStrings()	1328
12.324.5.6	GetUIDName()	1328
12.324.5.7	GetUIDString()	1328
12.324.5.8	operator TSType()	1328
12.324.5.9	SetFromUID()	1329
12.325	gdcm::network::ULAction Class Reference	1329
12.325.1	Detailed Description	1331
12.325.2	Constructor & Destructor Documentation	1331
12.325.2.1	ULAction() [1/2]	1331
12.325.2.2	~ULAction()	1331
12.325.2.3	ULAction() [2/2]	1332
12.325.3	Member Function Documentation	1332
12.325.3.1	operator=()	1332
12.325.3.2	PerformAction()	1332
12.326	gdcm::network::ULActionAA1 Class Reference	1333
12.326.1	Member Function Documentation	1334
12.326.1.1	PerformAction()	1334
12.327	gdcm::network::ULActionAA2 Class Reference	1334
12.327.1	Member Function Documentation	1335
12.327.1.1	PerformAction()	1335
12.328	gdcm::network::ULActionAA3 Class Reference	1335
12.328.1	Member Function Documentation	1336
12.328.1.1	PerformAction()	1336
12.329	gdcm::network::ULActionAA4 Class Reference	1337
12.329.1	Member Function Documentation	1338
12.329.1.1	PerformAction()	1338
12.330	gdcm::network::ULActionAA5 Class Reference	1338
12.330.1	Member Function Documentation	1339
12.330.1.1	PerformAction()	1339
12.331	gdcm::network::ULActionAA6 Class Reference	1339
12.331.1	Member Function Documentation	1340
12.331.1.1	PerformAction()	1340
12.332	gdcm::network::ULActionAA7 Class Reference	1341
12.332.1	Member Function Documentation	1342
12.332.1.1	PerformAction()	1342
12.333	gdcm::network::ULActionAA8 Class Reference	1342
12.333.1	Member Function Documentation	1343

12.333.1.1 PerformAction()	1343
12.334 gdcmm::network::ULActionAE1 Class Reference	1343
12.334.1 Member Function Documentation	1344
12.334.1.1 PerformAction()	1344
12.335 gdcmm::network::ULActionAE2 Class Reference	1345
12.335.1 Member Function Documentation	1346
12.335.1.1 PerformAction()	1346
12.336 gdcmm::network::ULActionAE3 Class Reference	1346
12.336.1 Member Function Documentation	1347
12.336.1.1 PerformAction()	1347
12.337 gdcmm::network::ULActionAE4 Class Reference	1347
12.337.1 Member Function Documentation	1348
12.337.1.1 PerformAction()	1348
12.338 gdcmm::network::ULActionAE5 Class Reference	1349
12.338.1 Member Function Documentation	1350
12.338.1.1 PerformAction()	1350
12.339 gdcmm::network::ULActionAE6 Class Reference	1350
12.339.1 Member Function Documentation	1351
12.339.1.1 PerformAction()	1351
12.340 gdcmm::network::ULActionAE7 Class Reference	1351
12.340.1 Member Function Documentation	1352
12.340.1.1 PerformAction()	1352
12.341 gdcmm::network::ULActionAE8 Class Reference	1353
12.341.1 Member Function Documentation	1354
12.341.1.1 PerformAction()	1354
12.342 gdcmm::network::ULActionAR1 Class Reference	1354
12.342.1 Member Function Documentation	1355
12.342.1.1 PerformAction()	1355
12.343 gdcmm::network::ULActionAR10 Class Reference	1355
12.343.1 Member Function Documentation	1356
12.343.1.1 PerformAction()	1356
12.344 gdcmm::network::ULActionAR2 Class Reference	1357
12.344.1 Member Function Documentation	1358
12.344.1.1 PerformAction()	1358
12.345 gdcmm::network::ULActionAR3 Class Reference	1358
12.345.1 Member Function Documentation	1359
12.345.1.1 PerformAction()	1359
12.346 gdcmm::network::ULActionAR4 Class Reference	1359
12.346.1 Member Function Documentation	1360
12.346.1.1 PerformAction()	1360

12.347	gdcmm::network::ULActionAR5 Class Reference	1361
12.347.1	Member Function Documentation	1362
12.347.1.1	PerformAction()	1362
12.348	gdcmm::network::ULActionAR6 Class Reference	1362
12.348.1	Member Function Documentation	1363
12.348.1.1	PerformAction()	1363
12.349	gdcmm::network::ULActionAR7 Class Reference	1363
12.349.1	Member Function Documentation	1364
12.349.1.1	PerformAction()	1364
12.350	gdcmm::network::ULActionAR8 Class Reference	1365
12.350.1	Member Function Documentation	1366
12.350.1.1	PerformAction()	1366
12.351	gdcmm::network::ULActionAR9 Class Reference	1366
12.351.1	Member Function Documentation	1367
12.351.1.1	PerformAction()	1367
12.352	gdcmm::network::ULActionDT1 Class Reference	1367
12.352.1	Member Function Documentation	1368
12.352.1.1	PerformAction()	1368
12.353	gdcmm::network::ULActionDT2 Class Reference	1369
12.353.1	Member Function Documentation	1370
12.353.1.1	PerformAction()	1370
12.354	gdcmm::network::ULBasicCallback Class Reference	1370
12.354.1	Detailed Description	1372
12.354.2	Constructor & Destructor Documentation	1372
12.354.2.1	ULBasicCallback()	1372
12.354.2.2	~ULBasicCallback()	1372
12.354.3	Member Function Documentation	1372
12.354.3.1	GetDataSets()	1372
12.354.3.2	GetResponses()	1372
12.354.3.3	HandleDataSet()	1372
12.354.3.4	HandleResponse()	1372
12.355	gdcmm::network::ULConnection Class Reference	1373
12.355.1	Detailed Description	1374
12.355.2	Constructor & Destructor Documentation	1374
12.355.2.1	ULConnection() [1/2]	1374
12.355.2.2	~ULConnection()	1374
12.355.2.3	ULConnection() [2/2]	1374
12.355.3	Member Function Documentation	1374
12.355.3.1	AddAcceptedPresentationContext()	1374
12.355.3.2	FindContext()	1375

12.355.3.3	GetAcceptedPresentationContexts() [1/2]	1375
12.355.3.4	GetAcceptedPresentationContexts() [2/2]	1375
12.355.3.5	GetConnectionInfo()	1375
12.355.3.6	GetMaxPDUSize()	1375
12.355.3.7	GetPresentationContextACByID()	1375
12.355.3.8	GetPresentationContextIDFromPresentationContext()	1375
12.355.3.9	GetPresentationContextRQByID()	1375
12.355.3.10	GetPresentationContexts()	1375
12.355.3.11	GetProtocol()	1376
12.355.3.12	GetState()	1376
12.355.3.13	GetTimer()	1376
12.355.3.14	InitializeConnection()	1376
12.355.3.15	InitializeIncomingConnection()	1376
12.355.3.16	operator=()	1376
12.355.3.17	SetMaxPDUSize()	1376
12.355.3.18	SetPresentationContexts() [1/2]	1376
12.355.3.19	SetPresentationContexts() [2/2]	1377
12.355.3.20	SetState()	1377
12.355.3.21	StopProtocol()	1377
12.355.4	Friends And Related Symbol Documentation	1377
12.355.4.1	ULActionAE6	1377
12.355.4.2	ULConnectionManager	1377
12.356	gdcmm::network::ULConnectionCallback Class Reference	1378
12.356.1	Detailed Description	1378
12.356.2	Constructor & Destructor Documentation	1379
12.356.2.1	ULConnectionCallback()	1379
12.356.2.2	~ULConnectionCallback()	1379
12.356.3	Member Function Documentation	1379
12.356.3.1	DataSetHandled()	1379
12.356.3.2	DataSetHandles()	1379
12.356.3.3	HandleDataSet()	1379
12.356.3.4	HandleResponse()	1379
12.356.3.5	ResetHandledDataSet()	1379
12.356.3.6	SetImplicitFlag()	1380
12.356.4	Member Data Documentation	1380
12.356.4.1	mImplicit	1380
12.357	gdcmm::network::ULConnectionInfo Class Reference	1380
12.357.1	Detailed Description	1380
12.357.2	Constructor & Destructor Documentation	1381
12.357.2.1	ULConnectionInfo()	1381

12.357.3 Member Function Documentation	1381
12.357.3.1 GetCalledAETitle()	1381
12.357.3.2 GetCalledComputerName()	1381
12.357.3.3 GetCalledIPAddress()	1381
12.357.3.4 GetCalledIPPort()	1381
12.357.3.5 GetCallingAETitle()	1381
12.357.3.6 GetMaxPDULength()	1381
12.357.3.7 Initialize()	1381
12.357.3.8 SetMaxPDULength()	1382
12.358 gdcmm::network::ULConnectionManager Class Reference	1382
12.358.1 Detailed Description	1385
12.358.2 Constructor & Destructor Documentation	1385
12.358.2.1 ULConnectionManager() [1/2]	1385
12.358.2.2 ULConnectionManager() [2/2]	1385
12.358.2.3 ~ULConnectionManager()	1385
12.358.3 Member Function Documentation	1385
12.358.3.1 BreakConnection()	1385
12.358.3.2 BreakConnectionNow()	1385
12.358.3.3 EstablishConnection()	1386
12.358.3.4 EstablishConnectionMove()	1386
12.358.3.5 RunEventLoop()	1386
12.358.3.6 RunMoveEventLoop()	1386
12.358.3.7 SendEcho()	1386
12.358.3.8 SendFind() [1/2]	1387
12.358.3.9 SendFind() [2/2]	1387
12.358.3.10 SendMove() [1/2]	1387
12.358.3.11 SendMove() [2/2]	1387
12.358.3.12 SendNAction() [1/2]	1387
12.358.3.13 SendNAction() [2/2]	1387
12.358.3.14 SendNCreate() [1/2]	1387
12.358.3.15 SendNCreate() [2/2]	1388
12.358.3.16 SendNDelete() [1/2]	1388
12.358.3.17 SendNDelete() [2/2]	1388
12.358.3.18 SendNEventReport() [1/2]	1388
12.358.3.19 SendNEventReport() [2/2]	1388
12.358.3.20 SendNGet() [1/2]	1388
12.358.3.21 SendNGet() [2/2]	1388
12.358.3.22 SendNSet() [1/2]	1388
12.358.3.23 SendNSet() [2/2]	1389
12.358.3.24 SendStore() [1/2]	1389

12.358.3.25 SendStore() [2/2]	1389
12.358.4 Member Data Documentation	1389
12.358.4.1 mConnection	1389
12.358.4.2 mSecondaryConnection	1389
12.358.4.3 mTransitions	1389
12.359 gdcm::network::ULEvent Class Reference	1390
12.359.1 Detailed Description	1390
12.359.2 Constructor & Destructor Documentation	1390
12.359.2.1 ULEvent() [1/2]	1390
12.359.2.2 ULEvent() [2/2]	1390
12.359.2.3 ~ULEvent()	1391
12.359.3 Member Function Documentation	1391
12.359.3.1 GetDataSetPos()	1391
12.359.3.2 GetEvent()	1391
12.359.3.3 GetIStream()	1391
12.359.3.4 GetPDUs()	1391
12.359.3.5 SetEvent()	1391
12.359.3.6 SetPDU()	1391
12.360 gdcm::network::ULTransitionTable Class Reference	1391
12.360.1 Detailed Description	1392
12.360.2 Constructor & Destructor Documentation	1392
12.360.2.1 ULTransitionTable()	1392
12.360.3 Member Function Documentation	1392
12.360.3.1 HandleEvent()	1392
12.360.3.2 PrintTable()	1392
12.361 gdcm::network::ULWritingCallback Class Reference	1393
12.361.1 Constructor & Destructor Documentation	1394
12.361.1.1 ULWritingCallback()	1394
12.361.1.2 ~ULWritingCallback()	1394
12.361.2 Member Function Documentation	1394
12.361.2.1 HandleDataSet()	1394
12.361.2.2 HandleResponse()	1394
12.361.2.3 SetDirectory()	1395
12.362 gdcm::UNExplicitDataElement Class Reference	1395
12.362.1 Detailed Description	1398
12.362.2 Member Function Documentation	1398
12.362.2.1 GetLength()	1398
12.362.2.2 Read()	1398
12.362.2.3 ReadPreValue()	1398
12.362.2.4 ReadValue()	1398

12.362.2.5 ReadWithLength()	1399
12.363 gdcmm::UNExplicitImplicitDataElement Class Reference	1399
12.363.1 Detailed Description	1402
12.363.2 Member Function Documentation	1402
12.363.2.1 GetLength()	1402
12.363.2.2 Read()	1402
12.363.2.3 ReadPreValue()	1402
12.363.2.4 ReadValue()	1402
12.364 gdcmm::Unpacker12Bits Class Reference	1403
12.364.1 Detailed Description	1403
12.364.2 Member Function Documentation	1403
12.364.2.1 Pack()	1403
12.364.2.2 Unpack()	1404
12.365 gdcmm::Usage Class Reference	1404
12.365.1 Detailed Description	1405
12.365.2 Member Enumeration Documentation	1405
12.365.2.1 UsageType	1405
12.365.3 Constructor & Destructor Documentation	1405
12.365.3.1 Usage()	1405
12.365.4 Member Function Documentation	1406
12.365.4.1 GetUsageString()	1406
12.365.4.2 GetUsageType()	1406
12.365.4.3 operator UsageType()	1406
12.365.5 Friends And Related Symbol Documentation	1406
12.365.5.1 operator<<	1406
12.366 gdcmm::UserEvent Class Reference	1407
12.367 gdcmm::network::UserInfoation Class Reference	1408
12.367.1 Detailed Description	1408
12.367.2 Constructor & Destructor Documentation	1409
12.367.2.1 UserInfoation() [1/2]	1409
12.367.2.2 ~UserInfoation()	1409
12.367.2.3 UserInfoation() [2/2]	1409
12.367.3 Member Function Documentation	1409
12.367.3.1 AddRoleSelectionSub()	1409
12.367.3.2 AddSOPClassExtendedNegociationSub()	1409
12.367.3.3 GetMaximumLengthSub() [1/2]	1409
12.367.3.4 GetMaximumLengthSub() [2/2]	1409
12.367.3.5 operator=()	1410
12.367.3.6 Print()	1410
12.367.3.7 Read()	1410

12.367.3.8 Size()	1410
12.367.3.9 Write()	1410
12.368 gdcm::UUIDGenerator Class Reference	1410
12.368.1 Detailed Description	1411
12.368.2 Member Function Documentation	1411
12.368.2.1 Generate()	1411
12.368.2.2 IsValid()	1411
12.369 gdcm::Validate Class Reference	1411
12.369.1 Detailed Description	1412
12.369.2 Constructor & Destructor Documentation	1412
12.369.2.1 Validate()	1412
12.369.2.2 ~Validate()	1413
12.369.3 Member Function Documentation	1413
12.369.3.1 GetValidatedFile()	1413
12.369.3.2 SetFile()	1413
12.369.3.3 Validation()	1413
12.369.4 Member Data Documentation	1413
12.369.4.1 F	1413
12.369.4.2 V	1413
12.370 gdcm::Value Class Reference	1414
12.370.1 Detailed Description	1415
12.370.2 Constructor & Destructor Documentation	1415
12.370.2.1 Value()	1415
12.370.2.2 ~Value()	1415
12.370.3 Member Function Documentation	1416
12.370.3.1 Clear()	1416
12.370.3.2 GetLength()	1416
12.370.3.3 operator==()	1416
12.370.3.4 SetLength()	1416
12.370.3.5 SetLengthOnly()	1416
12.370.4 Friends And Related Symbol Documentation	1417
12.370.4.1 DataElement	1417
12.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference	1417
12.371.1 Detailed Description	1417
12.371.2 Member Function Documentation	1417
12.371.2.1 Read()	1417
12.371.2.2 Write()	1418
12.372 gdcm::MrProtocol::Vector3 Struct Reference	1418
12.372.1 Member Data Documentation	1418
12.372.1.1 dCor	1418

12.372.1.2 dSag	1418
12.372.1.3 dTra	1418
12.373 gdcm::Version Class Reference	1419
12.373.1 Detailed Description	1419
12.373.2 Constructor & Destructor Documentation	1419
12.373.2.1 Version()	1419
12.373.2.2 ~Version()	1419
12.373.3 Member Function Documentation	1419
12.373.3.1 GetBuildVersion()	1419
12.373.3.2 GetMajorVersion()	1420
12.373.3.3 GetMinorVersion()	1420
12.373.3.4 GetVersion()	1420
12.373.3.5 Print()	1420
12.373.4 Friends And Related Symbol Documentation	1420
12.373.4.1 operator<<	1420
12.374 gdcm::VL Class Reference	1420
12.374.1 Detailed Description	1421
12.374.2 Member Typedef Documentation	1421
12.374.2.1 Type	1421
12.374.3 Constructor & Destructor Documentation	1422
12.374.3.1 VL()	1422
12.374.4 Member Function Documentation	1422
12.374.4.1 GetLength()	1422
12.374.4.2 GetVL16Max()	1422
12.374.4.3 GetVL32Max()	1422
12.374.4.4 IsOdd()	1422
12.374.4.5 IsUndefined()	1423
12.374.4.6 operator uint32_t()	1423
12.374.4.7 operator++() [1/2]	1423
12.374.4.8 operator++() [2/2]	1423
12.374.4.9 operator+=()	1423
12.374.4.10 Read()	1423
12.374.4.11 Read16()	1423
12.374.4.12 SetToUndefined()	1424
12.374.4.13 Write()	1424
12.374.4.14 Write16()	1424
12.374.5 Friends And Related Symbol Documentation	1424
12.374.5.1 operator<<	1424
12.375 gdcm::VM Class Reference	1424
12.375.1 Detailed Description	1426

12.375.2 Member Enumeration Documentation	1426
12.375.2.1 VMType	1426
12.375.3 Constructor & Destructor Documentation	1428
12.375.3.1 VM()	1428
12.375.4 Member Function Documentation	1428
12.375.4.1 Compatible()	1428
12.375.4.2 GetIndex()	1428
12.375.4.3 GetLength()	1428
12.375.4.4 GetNumberOfElementsFromArray()	1428
12.375.4.5 GetVMString()	1429
12.375.4.6 GetVMType()	1429
12.375.4.7 GetVMTypeFromLength()	1429
12.375.4.8 IsValid()	1429
12.375.4.9 operator VMType()	1429
12.375.5 Friends And Related Symbol Documentation	1429
12.375.5.1 operator<<	1429
12.376 gdcmm::VMToLength< T > Struct Template Reference	1430
12.377 gdcmm::VR Class Reference	1430
12.377.1 Detailed Description	1432
12.377.2 Member Enumeration Documentation	1432
12.377.2.1 VRType	1432
12.377.3 Constructor & Destructor Documentation	1433
12.377.3.1 VR()	1433
12.377.4 Member Function Documentation	1434
12.377.4.1 CanDisplay()	1434
12.377.4.2 Compatible()	1434
12.377.4.3 GetLength() [1/2]	1434
12.377.4.4 GetLength() [2/2]	1434
12.377.4.5 GetSize()	1434
12.377.4.6 GetSizeof()	1435
12.377.4.7 GetVRString()	1435
12.377.4.8 GetVRStringFromFile()	1435
12.377.4.9 GetVRType()	1435
12.377.4.10 GetVRTypeFromFile()	1435
12.377.4.11 IsASCII()	1435
12.377.4.12 IsASCII2()	1435
12.377.4.13 IsBinary()	1435
12.377.4.14 IsBinary2()	1436
12.377.4.15 IsDual()	1436
12.377.4.16 IsSwap()	1436

12.377.4.17 IsValid() [1/2]	1436
12.377.4.18 IsValid() [2/2]	1436
12.377.4.19 IsVRFile()	1436
12.377.4.20 operator VRType()	1436
12.377.4.21 Read()	1436
12.377.4.22 Write()	1437
12.377.5 Friends And Related Symbol Documentation	1437
12.377.5.1 operator<<	1437
12.378 gdcm::VR16ExplicitDataElement Class Reference	1437
12.378.1 Detailed Description	1440
12.378.2 Member Function Documentation	1440
12.378.2.1 GetLength()	1440
12.378.2.2 Read()	1440
12.378.2.3 ReadPreValue()	1440
12.378.2.4 ReadValue()	1440
12.378.2.5 ReadWithLength()	1441
12.379 gdcm::VRToEncoding< T > Struct Template Reference	1441
12.380 gdcm::VRToType< T > Struct Template Reference	1441
12.380.1 Detailed Description	1441
12.381 gdcm::VRVLSize< T > Class Template Reference	1442
12.382 gdcm::VRVLSize< 0 > Class Reference	1442
12.382.1 Member Function Documentation	1443
12.382.1.1 Read()	1443
12.382.1.2 Write()	1443
12.383 gdcm::VRVLSize< 1 > Class Reference	1444
12.383.1 Member Function Documentation	1444
12.383.1.1 Read()	1444
12.383.1.2 Write()	1445
12.384 vtkGDCMImageReader Class Reference	1445
12.384.1 Detailed Description	1447
12.384.2 Constructor & Destructor Documentation	1448
12.384.2.1 vtkGDCMImageReader()	1448
12.384.2.2 ~vtkGDCMImageReader()	1448
12.384.3 Member Function Documentation	1448
12.384.3.1 CanReadFile()	1448
12.384.3.2 ExecuteData()	1448
12.384.3.3 ExecuteInformation()	1448
12.384.3.4 FillMedicalImageInformation()	1449
12.384.3.5 GetDescriptiveName()	1449
12.384.3.6 GetFileExtensions()	1449

12.384.3.7	GetIconImage()	1449
12.384.3.8	GetOverlay()	1449
12.384.3.9	LoadSingleFile()	1449
12.384.3.10	New()	1449
12.384.3.11	PrintSelf()	1450
12.384.3.12	RequestDataCompat()	1450
12.384.3.13	RequestInformationCompat()	1450
12.384.3.14	SetCurve()	1450
12.384.3.15	SetFileNames()	1450
12.384.3.16	SetFilePattern()	1451
12.384.3.17	SetFilePrefix()	1451
12.384.3.18	SetMedicalImageProperties()	1451
12.384.3.19	vtkBooleanMacro() [1/5]	1451
12.384.3.20	vtkBooleanMacro() [2/5]	1451
12.384.3.21	vtkBooleanMacro() [3/5]	1451
12.384.3.22	vtkBooleanMacro() [4/5]	1451
12.384.3.23	vtkBooleanMacro() [5/5]	1452
12.384.3.24	vtkGetMacro() [1/11]	1452
12.384.3.25	vtkGetMacro() [2/11]	1452
12.384.3.26	vtkGetMacro() [3/11]	1452
12.384.3.27	vtkGetMacro() [4/11]	1452
12.384.3.28	vtkGetMacro() [5/11]	1452
12.384.3.29	vtkGetMacro() [6/11]	1453
12.384.3.30	vtkGetMacro() [7/11]	1453
12.384.3.31	vtkGetMacro() [8/11]	1453
12.384.3.32	vtkGetMacro() [9/11]	1453
12.384.3.33	vtkGetMacro() [10/11]	1453
12.384.3.34	vtkGetMacro() [11/11]	1453
12.384.3.35	vtkGetObjectMacro() [1/4]	1454
12.384.3.36	vtkGetObjectMacro() [2/4]	1454
12.384.3.37	vtkGetObjectMacro() [3/4]	1454
12.384.3.38	vtkGetObjectMacro() [4/4]	1454
12.384.3.39	vtkGetStringMacro() [1/2]	1454
12.384.3.40	vtkGetStringMacro() [2/2]	1454
12.384.3.41	vtkGetVector3Macro()	1455
12.384.3.42	vtkGetVector6Macro()	1455
12.384.3.43	vtkSetMacro() [1/4]	1455
12.384.3.44	vtkSetMacro() [2/4]	1455
12.384.3.45	vtkSetMacro() [3/4]	1455
12.384.3.46	vtkSetMacro() [4/4]	1456

12.384.3.47	vtkSetVector6Macro()	1456
12.384.3.48	vtkTypeMacro()	1456
12.384.4	Member Data Documentation	1456
12.384.4.1	ApplyInverseVideo	1456
12.384.4.2	ApplyLookupTable	1456
12.384.4.3	ApplyPlanarConfiguration	1456
12.384.4.4	ApplyShiftScale	1457
12.384.4.5	ApplyYBRToRGB	1457
12.384.4.6	Curve	1457
12.384.4.7	DirectionCosines	1457
12.384.4.8	FileNames	1457
12.384.4.9	ForceRescale	1457
12.384.4.10	IconDataScalarType	1457
12.384.4.11	IconImageDataExtent	1457
12.384.4.12	IconNumberOfScalarComponents	1458
12.384.4.13	ImageFormat	1458
12.384.4.14	ImageOrientationPatient	1458
12.384.4.15	ImagePositionPatient	1458
12.384.4.16	LoadIconImage	1458
12.384.4.17	LoadOverlays	1458
12.384.4.18	LossyFlag	1458
12.384.4.19	MedicalImageProperties	1459
12.384.4.20	NumberOfIconImages	1459
12.384.4.21	NumberOfOverlays	1459
12.384.4.22	PlanarConfiguration	1459
12.384.4.23	Scale	1459
12.384.4.24	Shift	1459
12.385	vtkGDCMImageReader2 Class Reference	1460
12.385.1	Detailed Description	1462
12.385.2	Constructor & Destructor Documentation	1462
12.385.2.1	vtkGDCMImageReader2()	1462
12.385.2.2	~vtkGDCMImageReader2()	1462
12.385.3	Member Function Documentation	1463
12.385.3.1	CanReadFile()	1463
12.385.3.2	FillMedicalImageInformation()	1463
12.385.3.3	GetDescriptiveName()	1463
12.385.3.4	GetFileExtensions()	1463
12.385.3.5	GetIconImage()	1463
12.385.3.6	GetIconImagePort()	1463
12.385.3.7	GetOverlay()	1463

12.385.3.8	GetOverlayPort()	1463
12.385.3.9	LoadSingleFile()	1464
12.385.3.10	New()	1464
12.385.3.11	PrintSelf()	1464
12.385.3.12	ProcessRequest()	1464
12.385.3.13	RequestData()	1464
12.385.3.14	RequestDataCompat()	1465
12.385.3.15	RequestInformation()	1465
12.385.3.16	RequestInformationCompat()	1465
12.385.3.17	SetCurve()	1465
12.385.3.18	SetFilePattern()	1465
12.385.3.19	SetFilePrefix()	1465
12.385.3.20	SetMedicalImageProperties()	1466
12.385.3.21	vtkBooleanMacro() [1/5]	1466
12.385.3.22	vtkBooleanMacro() [2/5]	1466
12.385.3.23	vtkBooleanMacro() [3/5]	1466
12.385.3.24	vtkBooleanMacro() [4/5]	1466
12.385.3.25	vtkBooleanMacro() [5/5]	1466
12.385.3.26	vtkGetMacro() [1/11]	1467
12.385.3.27	vtkGetMacro() [2/11]	1467
12.385.3.28	vtkGetMacro() [3/11]	1467
12.385.3.29	vtkGetMacro() [4/11]	1467
12.385.3.30	vtkGetMacro() [5/11]	1467
12.385.3.31	vtkGetMacro() [6/11]	1467
12.385.3.32	vtkGetMacro() [7/11]	1468
12.385.3.33	vtkGetMacro() [8/11]	1468
12.385.3.34	vtkGetMacro() [9/11]	1468
12.385.3.35	vtkGetMacro() [10/11]	1468
12.385.3.36	vtkGetMacro() [11/11]	1468
12.385.3.37	vtkGetObjectMacro() [1/2]	1468
12.385.3.38	vtkGetObjectMacro() [2/2]	1469
12.385.3.39	vtkGetStringMacro() [1/2]	1469
12.385.3.40	vtkGetStringMacro() [2/2]	1469
12.385.3.41	vtkGetVector3Macro()	1469
12.385.3.42	vtkGetVector6Macro()	1469
12.385.3.43	vtkSetMacro() [1/4]	1469
12.385.3.44	vtkSetMacro() [2/4]	1470
12.385.3.45	vtkSetMacro() [3/4]	1470
12.385.3.46	vtkSetMacro() [4/4]	1470
12.385.3.47	vtkSetVector6Macro()	1470

12.385.3.48	vtkTypeMacro()	1470
12.385.4	Member Data Documentation	1471
12.385.4.1	ApplyInverseVideo	1471
12.385.4.2	ApplyLookupTable	1471
12.385.4.3	ApplyPlanarConfiguration	1471
12.385.4.4	ApplyShiftScale	1471
12.385.4.5	ApplyYBRToRGB	1471
12.385.4.6	Curve	1471
12.385.4.7	DirectionCosines	1471
12.385.4.8	ForceRescale	1471
12.385.4.9	IconDataScalarType	1472
12.385.4.10	IconImageDataExtent	1472
12.385.4.11	IconNumberOfScalarComponents	1472
12.385.4.12	ImageFormat	1472
12.385.4.13	ImageOrientationPatient	1472
12.385.4.14	ImagePositionPatient	1472
12.385.4.15	LoadIconImage	1472
12.385.4.16	LoadOverlays	1472
12.385.4.17	LossyFlag	1473
12.385.4.18	NumberOfIconImages	1473
12.385.4.19	NumberOfOverlays	1473
12.385.4.20	PlanarConfiguration	1473
12.385.4.21	Scale	1473
12.385.4.22	Shift	1473
12.386	vtkGDCMImageWriter Class Reference	1474
12.386.1	Detailed Description	1476
12.386.2	Member Enumeration Documentation	1476
12.386.2.1	CompressionTypes	1476
12.386.3	Constructor & Destructor Documentation	1476
12.386.3.1	vtkGDCMImageWriter()	1476
12.386.3.2	~vtkGDCMImageWriter()	1476
12.386.4	Member Function Documentation	1476
12.386.4.1	GetDescriptiveName()	1476
12.386.4.2	GetFileExtensions()	1477
12.386.4.3	GetFileName()	1477
12.386.4.4	New()	1477
12.386.4.5	PrintSelf()	1477
12.386.4.6	SetDirectionCosines()	1477
12.386.4.7	SetDirectionCosinesFromImageOrientationPatient()	1477
12.386.4.8	SetFileNames()	1478

12.386.4.9 SetMedicalImageProperties()	1478
12.386.4.10 vtkBooleanMacro() [1/2]	1478
12.386.4.11 vtkBooleanMacro() [2/2]	1478
12.386.4.12 vtkGetMacro() [1/7]	1478
12.386.4.13 vtkGetMacro() [2/7]	1478
12.386.4.14 vtkGetMacro() [3/7]	1479
12.386.4.15 vtkGetMacro() [4/7]	1479
12.386.4.16 vtkGetMacro() [5/7]	1479
12.386.4.17 vtkGetMacro() [6/7]	1479
12.386.4.18 vtkGetMacro() [7/7]	1479
12.386.4.19 vtkGetObjectMacro() [1/3]	1479
12.386.4.20 vtkGetObjectMacro() [2/3]	1479
12.386.4.21 vtkGetObjectMacro() [3/3]	1480
12.386.4.22 vtkGetStringMacro() [1/2]	1480
12.386.4.23 vtkGetStringMacro() [2/2]	1480
12.386.4.24 vtkSetMacro() [1/7]	1480
12.386.4.25 vtkSetMacro() [2/7]	1480
12.386.4.26 vtkSetMacro() [3/7]	1480
12.386.4.27 vtkSetMacro() [4/7]	1480
12.386.4.28 vtkSetMacro() [5/7]	1481
12.386.4.29 vtkSetMacro() [6/7]	1481
12.386.4.30 vtkSetMacro() [7/7]	1481
12.386.4.31 vtkSetStringMacro() [1/2]	1481
12.386.4.32 vtkSetStringMacro() [2/2]	1481
12.386.4.33 vtkTypeMacro()	1481
12.386.4.34 Write()	1481
12.386.4.35 WriteGDCMData()	1482
12.386.4.36 WriteSlice()	1482
12.387 vtkGDCMMedicalImageProperties Class Reference	1482
12.387.1 Constructor & Destructor Documentation	1483
12.387.1.1 vtkGDCMMedicalImageProperties()	1483
12.387.1.2 ~vtkGDCMMedicalImageProperties()	1483
12.387.2 Member Function Documentation	1483
12.387.2.1 Clear()	1483
12.387.2.2 GetFile()	1484
12.387.2.3 New()	1484
12.387.2.4 PrintSelf()	1484
12.387.2.5 PushBackFile()	1484
12.387.2.6 vtkTypeMacro()	1484
12.387.3 Friends And Related Symbol Documentation	1484

12.387.3.1	vtkGDCMImageReader	1484
12.387.3.2	vtkGDCMImageReader2	1485
12.387.3.3	vtkGDCMImageWriter	1485
12.388	vtkGDCMPolyDataReader Class Reference	1485
12.388.1	Detailed Description	1487
12.388.2	Constructor & Destructor Documentation	1487
12.388.2.1	vtkGDCMPolyDataReader()	1487
12.388.2.2	~vtkGDCMPolyDataReader()	1487
12.388.3	Member Function Documentation	1487
12.388.3.1	FillMedicalImageInformation()	1487
12.388.3.2	New()	1487
12.388.3.3	PrintSelf()	1488
12.388.3.4	RequestData()	1488
12.388.3.5	RequestData__HemodynamicWaveformStorage()	1488
12.388.3.6	RequestData__RTStructureSetStorage()	1488
12.388.3.7	RequestInformation()	1488
12.388.3.8	RequestInformation__HemodynamicWaveformStorage()	1488
12.388.3.9	RequestInformation__RTStructureSetStorage()	1488
12.388.3.10	vtkGetObjectMacro() [1/2]	1489
12.388.3.11	vtkGetObjectMacro() [2/2]	1489
12.388.3.12	vtkGetStringMacro()	1489
12.388.3.13	vtkSetStringMacro()	1489
12.388.3.14	vtkTypeMacro()	1489
12.388.4	Member Data Documentation	1489
12.388.4.1	FileName	1489
12.388.4.2	MedicalImageProperties	1490
12.388.4.3	RTStructSetProperties	1490
12.389	vtkGDCMPolyDataWriter Class Reference	1490
12.389.1	Detailed Description	1492
12.389.2	Constructor & Destructor Documentation	1492
12.389.2.1	vtkGDCMPolyDataWriter()	1492
12.389.2.2	~vtkGDCMPolyDataWriter()	1492
12.389.3	Member Function Documentation	1492
12.389.3.1	InitializeRTStructSet()	1492
12.389.3.2	New()	1492
12.389.3.3	PrintSelf()	1493
12.389.3.4	SetMedicalImageProperties()	1493
12.389.3.5	SetNumberOfInputPorts()	1493
12.389.3.6	SetRTStructSetProperties()	1493
12.389.3.7	vtkTypeMacro()	1493

12.389.3.8	WriteData()	1493
12.389.3.9	WriteRTSTRUCTData()	1494
12.389.3.10	WriteRTSTRUCTInfo()	1494
12.389.4	Member Data Documentation	1494
12.389.4.1	MedicalImageProperties	1494
12.389.4.2	RTStructSetProperties	1494
12.390	vtkGDCMTesting Class Reference	1494
12.390.1	Detailed Description	1495
12.390.2	Member Typedef Documentation	1496
12.390.2.1	MD5MetaImagesType	1496
12.390.3	Constructor & Destructor Documentation	1496
12.390.3.1	vtkGDCMTesting()	1496
12.390.3.2	~vtkGDCMTesting()	1496
12.390.4	Member Function Documentation	1496
12.390.4.1	GetGDCMDataRoot()	1496
12.390.4.2	GetMD5MetaImage()	1496
12.390.4.3	GetMHDMD5FromFile()	1496
12.390.4.4	GetNumberOfMD5MetaImages()	1497
12.390.4.5	GetRAWMD5FromFile()	1497
12.390.4.6	GetVTKDataRoot()	1497
12.390.4.7	New()	1497
12.390.4.8	PrintSelf()	1497
12.390.4.9	vtkTypeMacro()	1497
12.391	vtkGDCMThreadedImageReader Class Reference	1498
12.391.1	Constructor & Destructor Documentation	1501
12.391.1.1	vtkGDCMThreadedImageReader()	1501
12.391.1.2	~vtkGDCMThreadedImageReader()	1501
12.391.2	Member Function Documentation	1501
12.391.2.1	ExecuteData()	1501
12.391.2.2	ExecuteInformation()	1501
12.391.2.3	New()	1501
12.391.2.4	PrintSelf()	1502
12.391.2.5	ReadFiles()	1502
12.391.2.6	RequestDataCompat()	1502
12.391.2.7	vtkBooleanMacro()	1502
12.391.2.8	vtkGetMacro()	1502
12.391.2.9	vtkSetMacro() [1/3]	1502
12.391.2.10	vtkSetMacro() [2/3]	1502
12.391.2.11	vtkSetMacro() [3/3]	1503
12.391.2.12	vtkTypeMacro()	1503

12.392	vtkGDCMThreadedImageReader2 Class Reference	1503
12.392.1	Constructor & Destructor Documentation	1505
12.392.1.1	vtkGDCMThreadedImageReader2()	1505
12.392.1.2	~vtkGDCMThreadedImageReader2()	1505
12.392.2	Member Function Documentation	1505
12.392.2.1	GetFileName()	1505
12.392.2.2	New()	1505
12.392.2.3	PrintSelf()	1505
12.392.2.4	RequestInformation()	1505
12.392.2.5	SetFileName()	1505
12.392.2.6	SetFileNames()	1506
12.392.2.7	SplitExtent()	1506
12.392.2.8	ThreadedRequestData()	1506
12.392.2.9	vtkBooleanMacro() [1/3]	1506
12.392.2.10	vtkBooleanMacro() [2/3]	1506
12.392.2.11	vtkBooleanMacro() [3/3]	1506
12.392.2.12	vtkGetMacro() [1/8]	1507
12.392.2.13	vtkGetMacro() [2/8]	1507
12.392.2.14	vtkGetMacro() [3/8]	1507
12.392.2.15	vtkGetMacro() [4/8]	1507
12.392.2.16	vtkGetMacro() [5/8]	1507
12.392.2.17	vtkGetMacro() [6/8]	1507
12.392.2.18	vtkGetMacro() [7/8]	1507
12.392.2.19	vtkGetMacro() [8/8]	1508
12.392.2.20	vtkGetObjectMacro()	1508
12.392.2.21	vtkGetVector3Macro() [1/2]	1508
12.392.2.22	vtkGetVector3Macro() [2/2]	1508
12.392.2.23	vtkGetVector6Macro()	1508
12.392.2.24	vtkSetMacro() [1/7]	1508
12.392.2.25	vtkSetMacro() [2/7]	1508
12.392.2.26	vtkSetMacro() [3/7]	1509
12.392.2.27	vtkSetMacro() [4/7]	1509
12.392.2.28	vtkSetMacro() [5/7]	1509
12.392.2.29	vtkSetMacro() [6/7]	1509
12.392.2.30	vtkSetMacro() [7/7]	1509
12.392.2.31	vtkSetVector3Macro() [1/2]	1509
12.392.2.32	vtkSetVector3Macro() [2/2]	1509
12.392.2.33	vtkSetVector6Macro()	1510
12.392.2.34	vtkTypeMacro()	1510
12.393	vtkImageColorViewer Class Reference	1510

12.393.1 Detailed Description	1513
12.393.2 Member Enumeration Documentation	1513
12.393.2.1 anonymous enum	1513
12.393.3 Constructor & Destructor Documentation	1513
12.393.3.1 vtkImageColorViewer()	1513
12.393.3.2 ~vtkImageColorViewer()	1514
12.393.4 Member Function Documentation	1514
12.393.4.1 AddInput()	1514
12.393.4.2 AddInputConnection()	1514
12.393.4.3 GetColorLevel()	1514
12.393.4.4 GetColorWindow()	1514
12.393.4.5 GetInput()	1514
12.393.4.6 GetOffScreenRendering()	1514
12.393.4.7 GetOverlayVisibility()	1514
12.393.4.8 GetPosition()	1514
12.393.4.9 GetSize()	1515
12.393.4.10 GetSliceMax()	1515
12.393.4.11 GetSliceMin()	1515
12.393.4.12 GetSliceRange() [1/3]	1515
12.393.4.13 GetSliceRange() [2/3]	1515
12.393.4.14 GetSliceRange() [3/3]	1515
12.393.4.15 GetWindowName()	1515
12.393.4.16 InstallPipeline()	1515
12.393.4.17 New()	1516
12.393.4.18 PrintSelf()	1516
12.393.4.19 Render()	1516
12.393.4.20 SetColorLevel()	1516
12.393.4.21 SetColorWindow()	1516
12.393.4.22 SetDisplayId()	1516
12.393.4.23 SetInput()	1517
12.393.4.24 SetInputConnection()	1517
12.393.4.25 SetOffScreenRendering()	1517
12.393.4.26 SetOverlayVisibility()	1517
12.393.4.27 SetParentId()	1517
12.393.4.28 SetPosition() [1/2]	1517
12.393.4.29 SetPosition() [2/2]	1517
12.393.4.30 SetRenderer()	1518
12.393.4.31 SetRenderWindow()	1518
12.393.4.32 SetSize() [1/2]	1518
12.393.4.33 SetSize() [2/2]	1518

12.393.4.34	SetSlice()	1518
12.393.4.35	SetSliceOrientation()	1518
12.393.4.36	SetSliceOrientationToXY()	1519
12.393.4.37	SetSliceOrientationToXZ()	1519
12.393.4.38	SetSliceOrientationToYZ()	1519
12.393.4.39	SetupInteractor()	1519
12.393.4.40	SetWindowId()	1519
12.393.4.41	UnInstallPipeline()	1519
12.393.4.42	UpdateDisplayExtent()	1519
12.393.4.43	UpdateOrientation()	1520
12.393.4.44	vtkBooleanMacro()	1520
12.393.4.45	vtkGetMacro() [1/2]	1520
12.393.4.46	vtkGetMacro() [2/2]	1520
12.393.4.47	vtkGetObjectMacro() [1/5]	1520
12.393.4.48	vtkGetObjectMacro() [2/5]	1520
12.393.4.49	vtkGetObjectMacro() [3/5]	1521
12.393.4.50	vtkGetObjectMacro() [4/5]	1521
12.393.4.51	vtkGetObjectMacro() [5/5]	1521
12.393.4.52	vtkTypeMacro()	1521
12.393.5	Friends And Related Symbol Documentation	1521
12.393.5.1	vtkImageColorViewerCallback	1521
12.393.6	Member Data Documentation	1521
12.393.6.1	FirstRender	1521
12.393.6.2	ImageActor	1522
12.393.6.3	Interactor	1522
12.393.6.4	InteractorStyle	1522
12.393.6.5	OverlayImageActor	1522
12.393.6.6	Renderer	1522
12.393.6.7	RenderWindow	1522
12.393.6.8	Slice	1522
12.393.6.9	SliceOrientation	1522
12.393.6.10	WindowLevel	1523
12.394	vtkImageMapToColors16 Class Reference	1523
12.394.1	Constructor & Destructor Documentation	1524
12.394.1.1	vtkImageMapToColors16()	1524
12.394.1.2	~vtkImageMapToColors16()	1525
12.394.2	Member Function Documentation	1525
12.394.2.1	GetMTime()	1525
12.394.2.2	New()	1525
12.394.2.3	PrintSelf()	1525

12.394.2.4	RequestData()	1525
12.394.2.5	RequestInformation()	1525
12.394.2.6	SetLookupTable()	1525
12.394.2.7	SetOutputFormatToLuminance()	1526
12.394.2.8	SetOutputFormatToLuminanceAlpha()	1526
12.394.2.9	SetOutputFormatToRGB()	1526
12.394.2.10	SetOutputFormatToRGBA()	1526
12.394.2.11	ThreadedRequestData()	1526
12.394.2.12	vtkBooleanMacro()	1526
12.394.2.13	vtkGetMacro() [1/3]	1527
12.394.2.14	vtkGetMacro() [2/3]	1527
12.394.2.15	vtkGetMacro() [3/3]	1527
12.394.2.16	vtkGetObjectMacro()	1527
12.394.2.17	vtkSetMacro() [1/3]	1527
12.394.2.18	vtkSetMacro() [2/3]	1527
12.394.2.19	vtkSetMacro() [3/3]	1528
12.394.2.20	vtkTypeMacro()	1528
12.394.3	Member Data Documentation	1528
12.394.3.1	ActiveComponent	1528
12.394.3.2	DataWasPassed	1528
12.394.3.3	LookupTable	1528
12.394.3.4	OutputFormat	1528
12.394.3.5	PassAlphaToOutput	1529
12.395	vtkImageMapToWindowLevelColors2 Class Reference	1529
12.395.1	Constructor & Destructor Documentation	1530
12.395.1.1	vtkImageMapToWindowLevelColors2()	1530
12.395.1.2	~vtkImageMapToWindowLevelColors2()	1530
12.395.2	Member Function Documentation	1531
12.395.2.1	New()	1531
12.395.2.2	PrintSelf()	1531
12.395.2.3	RequestData()	1531
12.395.2.4	RequestInformation()	1531
12.395.2.5	ThreadedRequestData()	1531
12.395.2.6	vtkGetMacro() [1/2]	1531
12.395.2.7	vtkGetMacro() [2/2]	1532
12.395.2.8	vtkSetMacro() [1/2]	1532
12.395.2.9	vtkSetMacro() [2/2]	1532
12.395.2.10	vtkTypeMacro()	1532
12.395.3	Member Data Documentation	1532
12.395.3.1	Level	1532

12.395.3.2 Window	1532
12.396 vtkImagePlanarComponentsToComponents Class Reference	1533
12.396.1 Constructor & Destructor Documentation	1534
12.396.1.1 vtkImagePlanarComponentsToComponents()	1534
12.396.1.2 ~vtkImagePlanarComponentsToComponents()	1534
12.396.2 Member Function Documentation	1534
12.396.2.1 New()	1534
12.396.2.2 PrintSelf()	1534
12.396.2.3 RequestData()	1534
12.396.2.4 vtkTypeMacro()	1535
12.397 vtkImageRGBToYBR Class Reference	1535
12.397.1 Constructor & Destructor Documentation	1536
12.397.1.1 vtkImageRGBToYBR()	1536
12.397.1.2 ~vtkImageRGBToYBR()	1536
12.397.2 Member Function Documentation	1536
12.397.2.1 New()	1536
12.397.2.2 PrintSelf()	1536
12.397.2.3 ThreadedExecute()	1537
12.397.2.4 vtkTypeMacro()	1537
12.398 vtkImageYBRToRGB Class Reference	1537
12.398.1 Constructor & Destructor Documentation	1538
12.398.1.1 vtkImageYBRToRGB()	1538
12.398.1.2 ~vtkImageYBRToRGB()	1538
12.398.2 Member Function Documentation	1539
12.398.2.1 New()	1539
12.398.2.2 PrintSelf()	1539
12.398.2.3 ThreadedExecute()	1539
12.398.2.4 vtkTypeMacro()	1539
12.399 vtkLookupTable16 Class Reference	1540
12.399.1 Constructor & Destructor Documentation	1541
12.399.1.1 vtkLookupTable16()	1541
12.399.1.2 ~vtkLookupTable16()	1541
12.399.2 Member Function Documentation	1541
12.399.2.1 Build()	1541
12.399.2.2 GetPointer()	1541
12.399.2.3 MapScalarsThroughTable2()	1542
12.399.2.4 New()	1542
12.399.2.5 PrintSelf()	1542
12.399.2.6 SetNumberOfTableValues()	1542
12.399.2.7 vtkTypeMacro()	1542

12.399.2.8 WritePointer()	1542
12.399.3 Member Data Documentation	1543
12.399.3.1 Table16	1543
12.400 vtkRTStructSetProperties Class Reference	1543
12.400.1 Detailed Description	1545
12.400.2 Constructor & Destructor Documentation	1545
12.400.2.1 vtkRTStructSetProperties()	1545
12.400.2.2 ~vtkRTStructSetProperties()	1545
12.400.3 Member Function Documentation	1545
12.400.3.1 AddContourReferencedFrameOfReference()	1545
12.400.3.2 AddReferencedFrameOfReference()	1546
12.400.3.3 AddStructureSetROI()	1546
12.400.3.4 AddStructureSetROIObservation()	1546
12.400.3.5 Clear()	1546
12.400.3.6 DeepCopy()	1546
12.400.3.7 GetContourReferencedFrameOfReferenceClassUID()	1546
12.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID()	1546
12.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]	1547
12.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]	1547
12.400.3.11 GetNumberOfReferencedFrameOfReferences()	1547
12.400.3.12 GetNumberOfStructureSetROIs()	1547
12.400.3.13 GetReferencedFrameOfReferenceClassUID()	1547
12.400.3.14 GetReferencedFrameOfReferenceInstanceUID()	1547
12.400.3.15 GetStructureSetObservationNumber()	1547
12.400.3.16 GetStructureSetROIDescription()	1547
12.400.3.17 GetStructureSetROIGenerationAlgorithm()	1547
12.400.3.18 GetStructureSetROIName()	1548
12.400.3.19 GetStructureSetROINumber()	1548
12.400.3.20 GetStructureSetROIObservationLabel()	1548
12.400.3.21 GetStructureSetROIRefFrameRefUID()	1548
12.400.3.22 GetStructureSetRTROIInterpretedType()	1548
12.400.3.23 New()	1548
12.400.3.24 PrintSelf()	1548
12.400.3.25 vtkGetStringMacro() [1/9]	1549
12.400.3.26 vtkGetStringMacro() [2/9]	1549
12.400.3.27 vtkGetStringMacro() [3/9]	1549
12.400.3.28 vtkGetStringMacro() [4/9]	1549
12.400.3.29 vtkGetStringMacro() [5/9]	1549
12.400.3.30 vtkGetStringMacro() [6/9]	1549
12.400.3.31 vtkGetStringMacro() [7/9]	1550

12.400.3.32	vtkGetStringMacro() [8/9]	1550
12.400.3.33	vtkGetStringMacro() [9/9]	1550
12.400.3.34	vtkSetStringMacro() [1/9]	1550
12.400.3.35	vtkSetStringMacro() [2/9]	1550
12.400.3.36	vtkSetStringMacro() [3/9]	1550
12.400.3.37	vtkSetStringMacro() [4/9]	1551
12.400.3.38	vtkSetStringMacro() [5/9]	1551
12.400.3.39	vtkSetStringMacro() [6/9]	1551
12.400.3.40	vtkSetStringMacro() [7/9]	1551
12.400.3.41	vtkSetStringMacro() [8/9]	1551
12.400.3.42	vtkSetStringMacro() [9/9]	1551
12.400.3.43	vtkTypeMacro()	1552
12.400.4	Member Data Documentation	1552
12.400.4.1	Internals	1552
12.400.4.2	ReferenceFrameOfReferenceUID	1552
12.400.4.3	ReferenceSeriesInstanceUID	1552
12.400.4.4	SeriesInstanceUID	1552
12.400.4.5	SOPInstanceUID	1552
12.400.4.6	StructureSetDate	1552
12.400.4.7	StructureSetLabel	1553
12.400.4.8	StructureSetName	1553
12.400.4.9	StructureSetTime	1553
12.400.4.10	StudyInstanceUID	1553
12.401	gdcm::Waveform Class Reference	1553
12.401.1	Detailed Description	1553
12.401.2	Constructor & Destructor Documentation	1554
12.401.2.1	Waveform()	1554
12.402	gdcm::WLMFindQuery Class Reference	1554
12.402.1	Detailed Description	1557
12.402.2	Constructor & Destructor Documentation	1557
12.402.2.1	WLMFindQuery()	1557
12.402.3	Member Function Documentation	1557
12.402.3.1	GetAbstractSyntaxUID()	1557
12.402.3.2	GetTagListByLevel()	1557
12.402.3.3	GetValidDataSet()	1557
12.402.3.4	InitializeDataSet()	1557
12.402.3.5	ValidateQuery()	1558
12.402.4	Friends And Related Symbol Documentation	1558
12.402.4.1	QueryFactory	1558
12.403	gdcm::Writer Class Reference	1558

12.403.1 Detailed Description	1560
12.403.2 Constructor & Destructor Documentation	1561
12.403.2.1 Writer()	1561
12.403.2.2 ~Writer()	1561
12.403.3 Member Function Documentation	1561
12.403.3.1 CheckFileMetaInformationOff()	1561
12.403.3.2 CheckFileMetaInformationOn()	1561
12.403.3.3 GetCheckFileMetaInformation()	1561
12.403.3.4 GetFile()	1562
12.403.3.5 GetStreamPtr()	1562
12.403.3.6 SetCheckFileMetaInformation()	1562
12.403.3.7 SetFile()	1562
12.403.3.8 SetFileName()	1563
12.403.3.9 SetStream()	1563
12.403.3.10 SetWriteDataSetOnly()	1563
12.403.3.11 Write()	1563
12.403.4 Friends And Related Symbol Documentation	1564
12.403.4.1 StreamImageWriter	1564
12.403.5 Member Data Documentation	1564
12.403.5.1 Ofstream	1564
12.403.5.2 Stream	1564
12.404 gdcm::XMLDictReader Class Reference	1564
12.404.1 Detailed Description	1566
12.404.2 Constructor & Destructor Documentation	1566
12.404.2.1 XMLDictReader()	1566
12.404.2.2 ~XMLDictReader()	1566
12.404.3 Member Function Documentation	1566
12.404.3.1 CharacterDataHandler()	1566
12.404.3.2 EndElement()	1566
12.404.3.3 GetDict()	1566
12.404.3.4 HandleDescription()	1566
12.404.3.5 HandleEntry()	1567
12.404.3.6 StartElement()	1567
12.405 gdcm::XMLPrinter Class Reference	1567
12.405.1 Member Enumeration Documentation	1568
12.405.1.1 PrintStyles	1568
12.405.2 Constructor & Destructor Documentation	1568
12.405.2.1 XMLPrinter()	1568
12.405.2.2 ~XMLPrinter()	1568
12.405.3 Member Function Documentation	1569

12.405.3.1	GetPrintStyle()	1569
12.405.3.2	HandleBulkData()	1569
12.405.3.3	Print()	1569
12.405.3.4	PrintDataElement()	1569
12.405.3.5	PrintDataSet()	1569
12.405.3.6	PrintSQ()	1569
12.405.3.7	SetFile()	1570
12.405.3.8	SetStyle()	1570
12.405.4	Member Data Documentation	1570
12.405.4.1	F	1570
12.405.4.2	PrintStyle	1570
12.406	gdcm::XMLPrivateDictReader Class Reference	1571
12.406.1	Detailed Description	1572
12.406.2	Constructor & Destructor Documentation	1572
12.406.2.1	XMLPrivateDictReader()	1572
12.406.2.2	~XMLPrivateDictReader()	1572
12.406.3	Member Function Documentation	1573
12.406.3.1	CharacterDataHandler()	1573
12.406.3.2	EndElement()	1573
12.406.3.3	GetPrivateDict()	1573
12.406.3.4	HandleDescription()	1573
12.406.3.5	HandleEntry()	1573
12.406.3.6	StartElement()	1573
13	File Documentation	1575
13.1	README.txt File Reference	1575
13.2	TestsList.txt File Reference	1575
13.3	gdcmASN1.h File Reference	1575
13.4	gdcmASN1.h	1576
13.5	gdcmBase64.h File Reference	1577
13.6	gdcmBase64.h	1577
13.7	gdcmBoxRegion.h File Reference	1578
13.8	gdcmBoxRegion.h	1579
13.9	gdcmByteSwap.h File Reference	1579
13.10	gdcmByteSwap.h	1580
13.11	gdcmCAPICryptoFactory.h File Reference	1581
13.12	gdcmCAPICryptoFactory.h	1582
13.13	gdcmCAPICryptographicMessageSyntax.h File Reference	1582
13.14	gdcmCAPICryptographicMessageSyntax.h	1583
13.15	gdcmCommand.h File Reference	1585

13.16	gdcMCommand.h	1586
13.17	gdcMCryptoFactory.h File Reference	1588
13.18	gdcMCryptoFactory.h	1589
13.19	gdcMCryptographicMessageSyntax.h File Reference	1590
13.20	gdcMCryptographicMessageSyntax.h	1591
13.21	gdcMDataEvent.h File Reference	1592
13.22	gdcMDataEvent.h	1593
13.23	gdcMDeflateStream.h File Reference	1594
13.24	gdcMDeflateStream.h	1594
13.25	gdcMDirectory.h File Reference	1594
13.26	gdcMDirectory.h	1595
13.27	gdcMDummyValueGenerator.h File Reference	1597
13.28	gdcMDummyValueGenerator.h	1597
13.29	gdcMEvent.h File Reference	1598
13.29.1	Macro Definition Documentation	1599
13.29.1.1	gdcMEventMacro	1599
13.30	gdcMEvent.h	1600
13.31	gdcMException.h File Reference	1601
13.31.1	Macro Definition Documentation	1602
13.31.1.1	gdcM_assert	1602
13.31.1.2	gdcM_debug_assert	1603
13.31.1.3	gdcM_forced_assert	1603
13.32	gdcMException.h	1603
13.33	gdcMFilename.h File Reference	1605
13.34	gdcMFilename.h	1605
13.35	gdcMFileNameEvent.h File Reference	1606
13.36	gdcMFileNameEvent.h	1607
13.37	gdcMFilenameGenerator.h File Reference	1608
13.38	gdcMFilenameGenerator.h	1609
13.39	gdcMLegacyMacro.h File Reference	1609
13.39.1	Macro Definition Documentation	1610
13.39.1.1	GDCM_LEGACY	1610
13.39.1.2	GDCM_LEGACY_BODY	1610
13.39.1.3	GDCM_LEGACY_REPLACED_BODY	1611
13.39.1.4	GDCM_NOOP_STATEMENT	1611
13.40	gdcMLegacyMacro.h	1611
13.41	gdcMD5.h File Reference	1612
13.42	gdcMD5.h	1613
13.43	gdcMObject.h File Reference	1613
13.44	gdcMObject.h	1614

13.45	gdcmOpenSSLCryptoFactory.h File Reference	1616
13.46	gdcmOpenSSLCryptoFactory.h	1617
13.47	gdcmOpenSSLCryptographicMessageSyntax.h File Reference	1617
13.48	gdcmOpenSSLCryptographicMessageSyntax.h	1619
13.49	gdcmOpenSSLP7CryptoFactory.h File Reference	1619
13.50	gdcmOpenSSLP7CryptoFactory.h	1620
13.51	gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference	1621
13.52	gdcmOpenSSLP7CryptographicMessageSyntax.h	1622
13.53	gdcmProgressEvent.h File Reference	1623
13.54	gdcmProgressEvent.h	1624
13.55	gdcmRegion.h File Reference	1624
13.56	gdcmRegion.h	1626
13.57	gdcmSHA1.h File Reference	1627
13.58	gdcmSHA1.h	1627
13.59	gdcmSmartPointer.h File Reference	1628
13.60	gdcmSmartPointer.h	1629
13.61	gdcmStaticAssert.h File Reference	1630
13.61.1	Macro Definition Documentation	1631
13.61.1.1	GDCM_DO_JOIN	1631
13.61.1.2	GDCM_DO_JOIN2	1631
13.61.1.3	GDCM_JOIN	1631
13.61.1.4	GDCM_STATIC_ASSERT	1631
13.62	gdcmStaticAssert.h	1632
13.63	gdcmString.h File Reference	1632
13.64	gdcmString.h	1634
13.65	gdcmSubject.h File Reference	1636
13.66	gdcmSubject.h	1636
13.67	gdcmSwapCode.h File Reference	1637
13.68	gdcmSwapCode.h	1638
13.69	gdcmSwapper.h File Reference	1639
13.70	gdcmSwapper.h	1640
13.71	gdcmSystem.h File Reference	1642
13.72	gdcmSystem.h	1642
13.73	gdcmTerminal.h File Reference	1644
13.74	gdcmTerminal.h	1645
13.75	gdcmTestDriver.h File Reference	1646
13.76	gdcmTestDriver.h	1646
13.77	gdcmTesting.h File Reference	1647
13.78	gdcmTesting.h	1647
13.79	gdcmTrace.h File Reference	1649

13.79.1 Macro Definition Documentation	1650
13.79.1.1 GDCM_FUNCTION	1650
13.79.1.2 gdcmAssertAlwaysMacro	1650
13.79.1.3 gdcmAssertMacro	1651
13.79.1.4 gdcmDebugMacro	1651
13.79.1.5 gdcmErrorMacro	1652
13.79.1.6 gdcmWarningMacro	1652
13.80 gdcmTrace.h	1653
13.81 gdcmTypes.h File Reference	1655
13.82 gdcmTypes.h	1655
13.83 gdcmUnpacker12Bits.h File Reference	1656
13.84 gdcmUnpacker12Bits.h	1657
13.85 gdcmVersion.h File Reference	1657
13.86 gdcmVersion.h	1658
13.87 gdcmWin32.h File Reference	1659
13.87.1 Macro Definition Documentation	1659
13.87.1.1 GDCM_EXPORT	1659
13.88 gdcmWin32.h	1659
13.89 gdcmCSAHeaderDict.h File Reference	1660
13.90 gdcmCSAHeaderDict.h	1662
13.91 gdcmCSAHeaderDictEntry.h File Reference	1663
13.92 gdcmCSAHeaderDictEntry.h	1664
13.93 gdcmDict.h File Reference	1666
13.94 gdcmDict.h	1667
13.95 gdcmDictConverter.h File Reference	1671
13.96 gdcmDictConverter.h	1672
13.97 gdcmDictEntry.h File Reference	1673
13.98 gdcmDictEntry.h	1674
13.99 gdcmDicts.h File Reference	1676
13.100 gdcmDicts.h	1677
13.101 gdcmGlobal.h File Reference	1679
13.102 gdcmGlobal.h	1680
13.103 gdcmGroupDict.h File Reference	1681
13.104 gdcmGroupDict.h	1682
13.105 gdcmSOPClassUIDToIOD.h File Reference	1683
13.106 gdcmSOPClassUIDToIOD.h	1683
13.107 gdcmUIDs.h File Reference	1684
13.108 gdcmUIDs.h	1685
13.109 gdcmAttribute.h File Reference	1697
13.110 gdcmAttribute.h	1699

13.111	gdcmBasicOffsetTable.h File Reference	1712
13.112	gdcmBasicOffsetTable.h	1713
13.113	gdcmByteBuffer.h File Reference	1714
13.114	gdcmByteBuffer.h	1715
13.115	gdcmByteSwapFilter.h File Reference	1717
13.116	gdcmByteSwapFilter.h	1718
13.117	gdcmByteValue.h File Reference	1718
13.118	gdcmByteValue.h	1719
13.119	gdcmCodeString.h File Reference	1723
13.120	gdcmCodeString.h	1723
13.121	gdcmCP246ExplicitDataElement.h File Reference	1725
13.122	gdcmCP246ExplicitDataElement.h	1725
13.123	gdcmCSAElement.h File Reference	1726
13.124	gdcmCSAElement.h	1727
13.125	gdcmCSAHeader.h File Reference	1729
13.126	gdcmCSAHeader.h	1730
13.127	gdcmDataElement.h File Reference	1732
13.128	gdcmDataElement.h	1733
13.129	gdcmDataSet.h File Reference	1735
13.130	gdcmDataSet.h	1737
13.131	gdcmDataSetEvent.h File Reference	1740
13.132	gdcmDataSetEvent.h	1741
13.133	gdcmElement.h File Reference	1742
13.134	gdcmElement.h	1743
13.135	gdcmExplicitDataElement.h File Reference	1754
13.136	gdcmExplicitDataElement.h	1755
13.137	gdcmExplicitImplicitDataElement.h File Reference	1756
13.138	gdcmExplicitImplicitDataElement.h	1757
13.139	gdcmFile.h File Reference	1758
13.140	gdcmFile.h	1759
13.141	gdcmFileMetaInformation.h File Reference	1759
13.142	gdcmFileMetaInformation.h	1761
13.143	gdcmFileSet.h File Reference	1762
13.144	gdcmFileSet.h	1764
13.145	gdcmFragment.h File Reference	1764
13.146	gdcmFragment.h	1766
13.147	gdcmImplicitDataElement.h File Reference	1769
13.148	gdcmImplicitDataElement.h	1769
13.149	gdcmItem.h File Reference	1770
13.150	gdcmItem.h	1771

13.151	gdcmlO.h File Reference	1776
13.152	gdcmlO.h	1776
13.153	gdcmMediaStorage.h File Reference	1777
13.154	gdcmMediaStorage.h	1778
13.155	gdcmMrProtocol.h File Reference	1781
13.156	gdcmMrProtocol.h	1782
13.157	gdcmParseException.h File Reference	1783
13.158	gdcmParseException.h	1784
13.159	gdcmParser.h File Reference	1785
13.160	gdcmParser.h	1786
13.161	gdcmPDBelement.h File Reference	1788
13.162	gdcmPDBelement.h	1789
13.163	gdcmPDBHeader.h File Reference	1790
13.164	gdcmPDBHeader.h	1791
13.165	gdcmPreamble.h File Reference	1792
13.166	gdcmPreamble.h	1793
13.167	gdcmPrivateTag.h File Reference	1794
13.168	gdcmPrivateTag.h	1795
13.169	gdcmReader.h File Reference	1796
13.170	gdcmReader.h	1797
13.171	gdcmSequenceOfFragments.h File Reference	1798
13.172	gdcmSequenceOfFragments.h	1799
13.173	gdcmSequenceOfItems.h File Reference	1803
13.174	gdcmSequenceOfItems.h	1804
13.175	gdcmTag.h File Reference	1807
13.176	gdcmTag.h	1809
13.177	gdcmTagToVR.h File Reference	1812
13.178	gdcmTagToVR.h	1812
13.179	gdcmTransferSyntax.h File Reference	1813
13.180	gdcmTransferSyntax.h	1814
13.181	gdcmUNExplicitDataElement.h File Reference	1815
13.182	gdcmUNExplicitDataElement.h	1816
13.183	gdcmUNExplicitImplicitDataElement.h File Reference	1817
13.184	gdcmUNExplicitImplicitDataElement.h	1818
13.185	gdcmValue.h File Reference	1818
13.186	gdcmValue.h	1819
13.187	gdcmValueIO.h File Reference	1820
13.188	gdcmValueIO.h	1821
13.189	gdcmVL.h File Reference	1821
13.190	gdcmVL.h	1822

13.191	gdcmmVM.h File Reference	1824
13.191.1	Macro Definition Documentation	1825
13.191.1.1	TYPETOLENGTH	1825
13.192	gdcmmVM.h	1826
13.193	gdcmmVR.h File Reference	1827
13.193.1	Macro Definition Documentation	1829
13.193.1.1	TYPETOENCODING	1829
13.193.1.2	VRTypeTemplateCase	1829
13.194	gdcmmVR.h	1830
13.195	gdcmmVR16ExplicitDataElement.h File Reference	1834
13.196	gdcmmVR16ExplicitDataElement.h	1835
13.197	gdcmmWriter.h File Reference	1836
13.198	gdcmmWriter.h	1837
13.199	gdcmmDefinedTerms.h File Reference	1838
13.200	gdcmmDefinedTerms.h	1839
13.201	gdcmmDefs.h File Reference	1839
13.202	gdcmmDefs.h	1841
13.203	gdcmmEnumeratedValues.h File Reference	1842
13.204	gdcmmEnumeratedValues.h	1842
13.205	gdcmmIOD.h File Reference	1843
13.206	gdcmmIOD.h	1844
13.207	gdcmmIODEntry.h File Reference	1846
13.208	gdcmmIODEntry.h	1848
13.209	gdcmmIODs.h File Reference	1848
13.210	gdcmmIODs.h	1850
13.211	gdcmmMacro.h File Reference	1851
13.212	gdcmmMacro.h	1852
13.213	gdcmmMacroEntry.h File Reference	1854
13.213.1	Macro Definition Documentation	1855
13.213.1.1	GDCMMACROENTRY_H	1855
13.214	gdcmmMacroEntry.h	1856
13.215	gdcmmMacros.h File Reference	1857
13.216	gdcmmMacros.h	1858
13.217	gdcmmModule.h File Reference	1859
13.218	gdcmmModule.h	1861
13.219	gdcmmModuleEntry.h File Reference	1862
13.220	gdcmmModuleEntry.h	1864
13.221	gdcmmModules.h File Reference	1865
13.222	gdcmmModules.h	1866
13.223	gdcmmNestedModuleEntries.h File Reference	1867

13.224	gdcmNestedModuleEntries.h	1869
13.225	gdcmPatient.h File Reference	1869
13.226	gdcmPatient.h	1870
13.227	gdcmSeries.h File Reference	1871
13.228	gdcmSeries.h	1872
13.229	gdcmStudy.h File Reference	1873
13.230	gdcmStudy.h	1874
13.231	gdcmTable.h File Reference	1874
13.232	gdcmTable.h	1875
13.233	gdcmTableEntry.h File Reference	1876
13.234	gdcmTableEntry.h	1878
13.235	gdcmTableReader.h File Reference	1878
13.236	gdcmTableReader.h	1880
13.237	gdcmType.h File Reference	1881
13.238	gdcmType.h	1882
13.239	gdcmUsage.h File Reference	1883
13.240	gdcmUsage.h	1886
13.241	gdcmXMLDictReader.h File Reference	1886
13.242	gdcmXMLDictReader.h	1887
13.243	gdcmXMLPrivateDictReader.h File Reference	1888
13.244	gdcmXMLPrivateDictReader.h	1889
13.245	gdcmAnonymizeEvent.h File Reference	1889
13.246	gdcmAnonymizeEvent.h	1891
13.247	gdcmAnonymizer.h File Reference	1891
13.248	gdcmAnonymizer.h	1892
13.249	gdcmApplicationEntity.h File Reference	1894
13.250	gdcmApplicationEntity.h	1894
13.251	gdcmAudioCodec.h File Reference	1895
13.252	gdcmAudioCodec.h	1896
13.253	gdcmBitmap.h File Reference	1896
13.254	gdcmBitmap.h	1897
13.255	gdcmBitmapToBitmapFilter.h File Reference	1900
13.256	gdcmBitmapToBitmapFilter.h	1900
13.257	gdcmCleaner.h File Reference	1901
13.258	gdcmCleaner.h	1902
13.259	gdcmCodec.h File Reference	1903
13.260	gdcmCodec.h	1904
13.261	gdcmCoder.h File Reference	1904
13.262	gdcmCoder.h	1905
13.263	gdcmConstCharWrapper.h File Reference	1906

13.264	gdcmlConstCharWrapper.h	1906
13.265	gdcmlCurve.h File Reference	1907
13.266	gdcmlCurve.h	1908
13.267	gdcmlDataSetHelper.h File Reference	1910
13.268	gdcmlDataSetHelper.h	1910
13.269	gdcmlDecoder.h File Reference	1911
13.270	gdcmlDecoder.h	1912
13.271	gdcmlDeltaEncodingCodec.h File Reference	1913
13.272	gdcmlDeltaEncodingCodec.h	1913
13.273	gdcmlDICOMDIR.h File Reference	1914
13.274	gdcmlDICOMDIR.h	1915
13.275	gdcmlDICOMDIRGenerator.h File Reference	1915
13.276	gdcmlDICOMDIRGenerator.h	1916
13.277	gdcmlDictPrinter.h File Reference	1917
13.278	gdcmlDictPrinter.h	1918
13.279	gdcmlDirectionCosines.h File Reference	1918
13.280	gdcmlDirectionCosines.h	1919
13.281	gdcmlDirectoryHelper.h File Reference	1920
13.282	gdcmlDirectoryHelper.h	1920
13.283	gdcmlDPath.h File Reference	1921
13.284	gdcmlDPath.h	1923
13.285	gdcmlDumper.h File Reference	1924
13.286	gdcmlDumper.h	1924
13.287	gdcmlEmptyMaskGenerator.h File Reference	1925
13.288	gdcmlEmptyMaskGenerator.h	1926
13.289	gdcmlEncapsulatedDocument.h File Reference	1927
13.290	gdcmlEncapsulatedDocument.h	1927
13.291	gdcmlEquipmentManufacturer.h File Reference	1928
13.292	gdcmlEquipmentManufacturer.h	1928
13.293	gdcmlFiducials.h File Reference	1929
13.294	gdcmlFiducials.h	1930
13.295	gdcmlFileAnonymizer.h File Reference	1930
13.296	gdcmlFileAnonymizer.h	1931
13.297	gdcmlFileChangeTransferSyntax.h File Reference	1932
13.298	gdcmlFileChangeTransferSyntax.h	1933
13.299	gdcmlFileDecompressLookupTable.h File Reference	1934
13.300	gdcmlFileDecompressLookupTable.h	1935
13.301	gdcmlFileDerivation.h File Reference	1936
13.302	gdcmlFileDerivation.h	1936
13.303	gdcmlFileExplicitFilter.h File Reference	1938

13.304	gdcmFileExplicitFilter.h	1938
13.305	gdcmFileStreamer.h File Reference	1939
13.306	gdcmFileStreamer.h	1940
13.307	gdcmIconImage.h File Reference	1941
13.308	gdcmIconImage.h	1942
13.309	gdcmIconImageFilter.h File Reference	1943
13.310	gdcmIconImageFilter.h	1944
13.311	gdcmIconImageGenerator.h File Reference	1945
13.312	gdcmIconImageGenerator.h	1946
13.313	gdcmImage.h File Reference	1946
13.314	gdcmImage.h	1948
13.315	gdcmImageApplyLookupTable.h File Reference	1949
13.316	gdcmImageApplyLookupTable.h	1949
13.317	gdcmImageChangePhotometricInterpretation.h File Reference	1950
13.318	gdcmImageChangePhotometricInterpretation.h	1951
13.319	gdcmImageChangePlanarConfiguration.h File Reference	1953
13.320	gdcmImageChangePlanarConfiguration.h	1953
13.321	gdcmImageChangeTransferSyntax.h File Reference	1954
13.322	gdcmImageChangeTransferSyntax.h	1955
13.323	gdcmImageCodec.h File Reference	1956
13.324	gdcmImageCodec.h	1957
13.325	gdcmImageConverter.h File Reference	1959
13.326	gdcmImageConverter.h	1960
13.327	gdcmImageFragmentSplitter.h File Reference	1961
13.328	gdcmImageFragmentSplitter.h	1961
13.329	gdcmImageHelper.h File Reference	1962
13.330	gdcmImageHelper.h	1963
13.331	gdcmImageReader.h File Reference	1964
13.332	gdcmImageReader.h	1965
13.333	gdcmImageRegionReader.h File Reference	1966
13.334	gdcmImageRegionReader.h	1967
13.335	gdcmImageToImageFilter.h File Reference	1968
13.336	gdcmImageToImageFilter.h	1969
13.337	gdcmImageWriter.h File Reference	1969
13.338	gdcmImageWriter.h	1970
13.339	gdcmIPPSorter.h File Reference	1971
13.340	gdcmIPPSorter.h	1972
13.341	gdcmJPEG12Codec.h File Reference	1973
13.342	gdcmJPEG12Codec.h	1973
13.343	gdcmJPEG16Codec.h File Reference	1974

13.344	gdcMJPEG16Codec.h	1975
13.345	gdcMJPEG2000Codec.h File Reference	1976
13.346	gdcMJPEG2000Codec.h	1976
13.347	gdcMJPEG8Codec.h File Reference	1978
13.348	gdcMJPEG8Codec.h	1978
13.349	gdcMJPEGCodec.h File Reference	1979
13.350	gdcMJPEGCodec.h	1980
13.351	gdcMJPEGLSCodec.h File Reference	1982
13.352	gdcMJPEGLSCodec.h	1982
13.353	gdcJSON.h File Reference	1983
13.354	gdcJSON.h	1984
13.355	gdcKAKADUCodec.h File Reference	1985
13.356	gdcKAKADUCodec.h	1986
13.357	gdcLookupTable.h File Reference	1986
13.358	gdcLookupTable.h	1987
13.359	gdcMEC_MR3.h File Reference	1989
13.360	gdcMEC_MR3.h	1989
13.361	gdcMeshPrimitive.h File Reference	1990
13.362	gdcMeshPrimitive.h	1991
13.363	gdcOrientation.h File Reference	1993
13.364	gdcOrientation.h	1993
13.365	gdcOverlay.h File Reference	1994
13.366	gdcOverlay.h	1995
13.367	gdcPDFCodec.h File Reference	1997
13.368	gdcPDFCodec.h	1997
13.369	gdcPersonName.h File Reference	1998
13.370	gdcPersonName.h	1999
13.371	gdcPGXCodec.h File Reference	2000
13.372	gdcPGXCodec.h	2001
13.373	gdcPhotometricInterpretation.h File Reference	2001
13.374	gdcPhotometricInterpretation.h	2003
13.375	gdcPixelFormat.h File Reference	2004
13.376	gdcPixelFormat.h	2005
13.377	gdcPixmap.h File Reference	2008
13.378	gdcPixmap.h	2009
13.379	gdcPixmapReader.h File Reference	2010
13.380	gdcPixmapReader.h	2011
13.381	gdcPixmapToPixmapFilter.h File Reference	2012
13.382	gdcPixmapToPixmapFilter.h	2013
13.383	gdcPixmapWriter.h File Reference	2013

13.384	gdcmPixmapWriter.h	2014
13.385	gdcmPNMCodec.h File Reference	2016
13.386	gdcmPNMCodec.h	2016
13.387	gdcmPrinter.h File Reference	2017
13.388	gdcmPrinter.h	2018
13.389	gdcmPVRGCodec.h File Reference	2020
13.390	gdcmPVRGCodec.h	2020
13.391	gdcmRAWCodec.h File Reference	2021
13.392	gdcmRAWCodec.h	2022
13.393	gdcmRescaler.h File Reference	2022
13.394	gdcmRescaler.h	2023
13.395	gdcmRLECodec.h File Reference	2025
13.396	gdcmRLECodec.h	2025
13.397	gdcmScanner.h File Reference	2026
13.398	gdcmScanner.h	2027
13.399	gdcmScanner2.h File Reference	2029
13.400	gdcmScanner2.h	2030
13.401	gdcmSegment.h File Reference	2032
13.402	gdcmSegment.h	2034
13.403	gdcmSegmentedPaletteColorLookupTable.h File Reference	2036
13.404	gdcmSegmentedPaletteColorLookupTable.h	2036
13.405	gdcmSegmentHelper.h File Reference	2037
13.406	gdcmSegmentHelper.h	2038
13.407	gdcmSegmentReader.h File Reference	2039
13.408	gdcmSegmentReader.h	2041
13.409	gdcmSegmentWriter.h File Reference	2041
13.410	gdcmSegmentWriter.h	2043
13.411	gdcmSerieHelper.h File Reference	2043
13.412	gdcmSerieHelper.h	2045
13.413	gdcmSimpleSubjectWatcher.h File Reference	2046
13.414	gdcmSimpleSubjectWatcher.h	2047
13.415	gdcmSorter.h File Reference	2048
13.416	gdcmSorter.h	2050
13.417	gdcmSpacing.h File Reference	2051
13.418	gdcmSpacing.h	2051
13.419	gdcmSpectroscopy.h File Reference	2053
13.420	gdcmSpectroscopy.h	2053
13.421	gdcmSplitMosaicFilter.h File Reference	2054
13.422	gdcmSplitMosaicFilter.h	2054
13.423	gdcmStreamImageReader.h File Reference	2056

13.424	gdcmStreamImageReader.h	2056
13.425	gdcmStreamImageWriter.h File Reference	2057
13.426	gdcmStreamImageWriter.h	2058
13.427	gdcmStrictScanner.h File Reference	2059
13.428	gdcmStrictScanner.h	2060
13.429	gdcmStrictScanner2.h File Reference	2062
13.430	gdcmStrictScanner2.h	2063
13.431	gdcmStringFilter.h File Reference	2065
13.432	gdcmStringFilter.h	2066
13.433	gdcmSurface.h File Reference	2066
13.434	gdcmSurface.h	2068
13.435	gdcmSurfaceHelper.h File Reference	2071
13.436	gdcmSurfaceHelper.h	2071
13.437	gdcmSurfaceReader.h File Reference	2073
13.438	gdcmSurfaceReader.h	2074
13.439	gdcmSurfaceWriter.h File Reference	2075
13.440	gdcmSurfaceWriter.h	2076
13.441	gdcmTagPath.h File Reference	2076
13.442	gdcmTagPath.h	2077
13.443	gdcmUIDGenerator.h File Reference	2078
13.444	gdcmUIDGenerator.h	2079
13.445	gdcmUUIDGenerator.h File Reference	2080
13.446	gdcmUUIDGenerator.h	2080
13.447	gdcmValidate.h File Reference	2081
13.448	gdcmValidate.h	2082
13.449	gdcmWaveform.h File Reference	2082
13.450	gdcmWaveform.h	2083
13.451	gdcmXMLPrinter.h File Reference	2083
13.452	gdcmXMLPrinter.h	2084
13.453	gdcmAAbortPDU.h File Reference	2086
13.454	gdcmAAbortPDU.h	2087
13.455	gdcmAAssociateACPDU.h File Reference	2087
13.456	gdcmAAssociateACPDU.h	2088
13.457	gdcmAAssociateRJPDU.h File Reference	2090
13.458	gdcmAAssociateRJPDU.h	2090
13.459	gdcmAAssociateRQPDU.h File Reference	2091
13.460	gdcmAAssociateRQPDU.h	2092
13.461	gdcmAbstractSyntax.h File Reference	2094
13.462	gdcmAbstractSyntax.h	2095
13.463	gdcmApplicationContext.h File Reference	2096

13.464	gdcApplicationContext.h	2097
13.465	gdcAReleaseRPPDU.h File Reference	2097
13.466	gdcAReleaseRPPDU.h	2098
13.467	gdcAReleaseRQPDU.h File Reference	2099
13.468	gdcAReleaseRQPDU.h	2100
13.469	gdcARTIMTimer.h File Reference	2101
13.470	gdcARTIMTimer.h	2101
13.471	gdcAsynchronousOperationsWindowSub.h File Reference	2102
13.472	gdcAsynchronousOperationsWindowSub.h	2103
13.473	gdcBaseCompositeMessage.h File Reference	2103
13.474	gdcBaseCompositeMessage.h	2104
13.475	gdcBaseNormalizedMessage.h File Reference	2105
13.476	gdcBaseNormalizedMessage.h	2106
13.477	gdcBasePDU.h File Reference	2107
13.478	gdcBasePDU.h	2108
13.479	gdcBaseQuery.h File Reference	2108
13.480	gdcBaseQuery.h	2110
13.481	gdcBaseRootQuery.h File Reference	2111
13.482	gdcBaseRootQuery.h	2112
13.483	gdcCEchoMessages.h File Reference	2113
13.484	gdcCEchoMessages.h	2114
13.485	gdcCFindMessages.h File Reference	2114
13.486	gdcCFindMessages.h	2115
13.487	gdcCMoveMessages.h File Reference	2116
13.488	gdcCMoveMessages.h	2117
13.489	gdcCommandDataSet.h File Reference	2118
13.490	gdcCommandDataSet.h	2118
13.491	gdcCompositeMessageFactory.h File Reference	2119
13.492	gdcCompositeMessageFactory.h	2120
13.493	gdcCompositeNetworkFunctions.h File Reference	2121
13.494	gdcCompositeNetworkFunctions.h	2122
13.495	gdcCStoreMessages.h File Reference	2123
13.496	gdcCStoreMessages.h	2123
13.497	gdcDIMSE.h File Reference	2124
13.498	gdcDIMSE.h	2125
13.499	gdcFindPatientRootQuery.h File Reference	2127
13.500	gdcFindPatientRootQuery.h	2128
13.501	gdcFindStudyRootQuery.h File Reference	2128
13.502	gdcFindStudyRootQuery.h	2129
13.503	gdcImplementationClassUIDSub.h File Reference	2130

13.504	gdcmImplementationClassUIDSub.h	2131
13.505	gdcmImplementationUIDSub.h File Reference	2132
13.506	gdcmImplementationUIDSub.h	2132
13.507	gdcmImplementationVersionNameSub.h File Reference	2133
13.508	gdcmImplementationVersionNameSub.h	2134
13.509	gdcmMaximumLengthSub.h File Reference	2135
13.510	gdcmMaximumLengthSub.h	2136
13.511	gdcmModalityPerformedProcedureStepCreateQuery.h File Reference	2137
13.512	gdcmModalityPerformedProcedureStepCreateQuery.h	2138
13.513	gdcmModalityPerformedProcedureStepSetQuery.h File Reference	2138
13.514	gdcmModalityPerformedProcedureStepSetQuery.h	2139
13.515	gdcmMovePatientRootQuery.h File Reference	2140
13.516	gdcmMovePatientRootQuery.h	2140
13.517	gdcmMoveStudyRootQuery.h File Reference	2141
13.518	gdcmMoveStudyRootQuery.h	2142
13.519	gdcmNActionMessages.h File Reference	2142
13.520	gdcmNActionMessages.h	2143
13.521	gdcmNCreateMessages.h File Reference	2144
13.522	gdcmNCreateMessages.h	2144
13.523	gdcmNDeleteMessages.h File Reference	2145
13.524	gdcmNDeleteMessages.h	2146
13.525	gdcmNetworkEvents.h File Reference	2146
13.526	gdcmNetworkEvents.h	2147
13.527	gdcmNetworkStateID.h File Reference	2148
13.528	gdcmNetworkStateID.h	2149
13.529	gdcmNEventReportMessages.h File Reference	2150
13.530	gdcmNEventReportMessages.h	2151
13.531	gdcmNGetMessages.h File Reference	2152
13.532	gdcmNGetMessages.h	2152
13.533	gdcmNormalizedMessageFactory.h File Reference	2153
13.534	gdcmNormalizedMessageFactory.h	2154
13.535	gdcmNormalizedNetworkFunctions.h File Reference	2154
13.536	gdcmNormalizedNetworkFunctions.h	2155
13.537	gdcmNSetMessages.h File Reference	2156
13.538	gdcmNSetMessages.h	2157
13.539	gdcmPDDataTFPDU.h File Reference	2157
13.540	gdcmPDDataTFPDU.h	2158
13.541	gdcmPDUFactory.h File Reference	2159
13.542	gdcmPDUFactory.h	2160
13.543	gdcmPresentationContext.h File Reference	2161

13.544	gdcmPresentationContext.h	2162
13.545	gdcmPresentationContextAC.h File Reference	2163
13.546	gdcmPresentationContextAC.h	2164
13.547	gdcmPresentationContextGenerator.h File Reference	2165
13.548	gdcmPresentationContextGenerator.h	2166
13.549	gdcmPresentationContextRQ.h File Reference	2166
13.550	gdcmPresentationContextRQ.h	2167
13.551	gdcmPresentationDataValue.h File Reference	2169
13.552	gdcmPresentationDataValue.h	2170
13.553	gdcmQueryBase.h File Reference	2171
13.554	gdcmQueryBase.h	2172
13.555	gdcmQueryFactory.h File Reference	2173
13.556	gdcmQueryFactory.h	2174
13.557	gdcmQueryImage.h File Reference	2175
13.558	gdcmQueryImage.h	2176
13.559	gdcmQueryPatient.h File Reference	2177
13.560	gdcmQueryPatient.h	2178
13.561	gdcmQuerySeries.h File Reference	2179
13.562	gdcmQuerySeries.h	2180
13.563	gdcmQueryStudy.h File Reference	2181
13.564	gdcmQueryStudy.h	2182
13.565	gdcmRoleSelectionSub.h File Reference	2182
13.566	gdcmRoleSelectionSub.h	2183
13.567	gdcmServiceClassApplicationInformation.h File Reference	2184
13.568	gdcmServiceClassApplicationInformation.h	2185
13.569	gdcmServiceClassUser.h File Reference	2185
13.570	gdcmServiceClassUser.h	2186
13.571	gdcmSOPClassExtendedNegociationSub.h File Reference	2188
13.572	gdcmSOPClassExtendedNegociationSub.h	2188
13.573	gdcmTransferSyntaxSub.h File Reference	2189
13.574	gdcmTransferSyntaxSub.h	2191
13.575	gdcmULAction.h File Reference	2191
13.576	gdcmULAction.h	2192
13.577	gdcmULActionAA.h File Reference	2193
13.578	gdcmULActionAA.h	2194
13.579	gdcmULActionAE.h File Reference	2195
13.580	gdcmULActionAE.h	2196
13.581	gdcmULActionAR.h File Reference	2197
13.582	gdcmULActionAR.h	2198
13.583	gdcmULActionDT.h File Reference	2200

13.584	gdcmlActionDT.h	2200
13.585	gdcmlBasicCallback.h File Reference	2201
13.586	gdcmlBasicCallback.h	2202
13.587	gdcmlConnection.h File Reference	2202
13.588	gdcmlConnection.h	2203
13.589	gdcmlConnectionCallback.h File Reference	2205
13.590	gdcmlConnectionCallback.h	2206
13.591	gdcmlConnectionInfo.h File Reference	2206
13.592	gdcmlConnectionInfo.h	2208
13.593	gdcmlConnectionManager.h File Reference	2208
13.594	gdcmlConnectionManager.h	2209
13.595	gdcmlEvent.h File Reference	2211
13.596	gdcmlEvent.h	2212
13.597	gdcmlTransitionTable.h File Reference	2213
13.598	gdcmlTransitionTable.h	2214
13.599	gdcmlWritingCallback.h File Reference	2216
13.600	gdcmlWritingCallback.h	2216
13.601	gdcmlUserInformation.h File Reference	2217
13.602	gdcmlUserInformation.h	2218
13.603	gdcmlWLMFindQuery.h File Reference	2219
13.604	gdcmlWLMFindQuery.h	2220
13.605	vtkGDCMImageReader.h File Reference	2220
13.605.1	Macro Definition Documentation	2222
13.605.1.1	VTK_CMYK	2222
13.605.1.2	VTK_INVERSE_LUMINANCE	2222
13.605.1.3	VTK_LOOKUP_TABLE	2222
13.605.1.4	VTK_YBR	2222
13.606	vtkGDCMImageReader.h	2222
13.607	vtkGDCMImageReader2.h File Reference	2226
13.607.1	Macro Definition Documentation	2227
13.607.1.1	VTK_CMYK	2227
13.607.1.2	VTK_INVERSE_LUMINANCE	2227
13.607.1.3	VTK_LOOKUP_TABLE	2227
13.607.1.4	VTK_YBR	2227
13.608	vtkGDCMImageReader2.h	2228
13.609	vtkGDCMImageWriter.h File Reference	2231
13.610	vtkGDCMImageWriter.h	2232
13.611	vtkGDCMMedicalImageProperties.h File Reference	2234
13.612	vtkGDCMMedicalImageProperties.h	2235
13.613	vtkGDCMPolyDataReader.h File Reference	2240

13.614	vtkGDCMPolyDataReader.h	2240
13.615	vtkGDCMPolyDataWriter.h File Reference	2241
13.616	vtkGDCMPolyDataWriter.h	2242
13.617	vtkGDCMTesting.h File Reference	2243
13.618	vtkGDCMTesting.h	2244
13.619	vtkGDCMThreadedImageReader.h File Reference	2245
13.620	vtkGDCMThreadedImageReader.h	2245
13.621	vtkGDCMThreadedImageReader2.h File Reference	2247
13.622	vtkGDCMThreadedImageReader2.h	2247
13.623	vtkImageColorViewer.h File Reference	2249
13.624	vtkImageColorViewer.h	2250
13.625	vtkImageMapToColors16.h File Reference	2253
13.626	vtkImageMapToColors16.h	2254
13.627	vtkImageMapToWindowLevelColors2.h File Reference	2255
13.628	vtkImageMapToWindowLevelColors2.h	2256
13.629	vtkImagePlanarComponentsToComponents.h File Reference	2257
13.630	vtkImagePlanarComponentsToComponents.h	2258
13.631	vtkImageRGBToYBR.h File Reference	2259
13.632	vtkImageRGBToYBR.h	2260
13.633	vtkImageYBRToRGB.h File Reference	2261
13.634	vtkImageYBRToRGB.h	2261
13.635	vtkLookupTable16.h File Reference	2262
13.636	vtkLookupTable16.h	2263
13.637	vtkRTStructSetProperties.h File Reference	2264
13.638	vtkRTStructSetProperties.h	2264
13.639	gdcmPythonFilter.h File Reference	2266
13.640	gdcmPythonFilter.h	2267
14	Examples	2269
14.1	TestByteSwap.cxx	2269
14.2	PatchFile.cxx	2271
14.3	SimplePrint.cs	2272
14.4	TestReader.cxx	2273
14.5	TestReader.py	2275
14.6	DecompressJPEGFile.cs	2275
14.7	ManipulateFile.cs	2276
14.8	ClinicalTrialIdentificationWorkflow.cs	2277
14.9	GenerateDICOMDIR.cs	2280
14.10	GenFakeImage.cxx	2281
14.11	ReformatFile.cs	2283

14.12 DecompressImage.cs	2284
14.13 StandardizeFiles.cs	2285
14.14 ScanDirectory.cs	2287
14.15 BasicAnonymizer.cs	2288
14.16 BasicImageAnonymizer.cs	2289
14.17 Cleaner.cs	2290
14.18 CompressLossyJPEG.cs	2292
14.19 DecompressImageMultiframe.cs	2293
14.20 DumpCSA.cs	2295
14.21 ExplicitLittleEndian.cs	2296
14.22 ExtractEncapsulatedFile.cs	2297
14.23 ExtractImageRegion.cs	2298
14.24 ExtractImageRegionWithLUT.cs	2300
14.25 ExtractOneFrame.cs	2301
14.26 FileAnonymize.cs	2302
14.27 FileChangeTS.cs	2303
14.28 FileChangeTSLossy.cs	2305
14.29 FileStreaming.cs	2308
14.30 GetArray.cs	2309
14.31 MpegVideoInfo.cs	2310
14.32 NewSequence.cs	2314
14.33 RescaleImage.cs	2315
14.34 SendFileSCU.cs	2316
14.35 SimplePrintPatientName.cs	2317
14.36 SortImage2.cs	2318
14.37 CStoreQtProgress.cxx	2318
14.38 ChangePrivateTags.cxx	2320
14.39 ChangeSequenceUltrasound.cxx	2322
14.40 CheckBigEndianBug.cxx	2323
14.41 ClinicalTrialAnnotate.cxx	2324
14.42 CompressImage.cxx	2325
14.43 ConvertToQImage.cxx	2326
14.44 CreateARGBImage.cxx	2328
14.45 CreateCMYKImage.cxx	2329
14.46 CreateJPIPDataSet.cxx	2330
14.47 DeriveSeries.cxx	2331
14.48 DiffFile.cxx	2332
14.49 DiscriminateVolume.cxx	2333
14.50 DumpADAC.cxx	2337
14.51 DumpExamCard.cxx	2342

14.52 DumpGEMSMovieGroup.cxx	2350
14.53 DumpImageHeaderInfo.cxx	2356
14.54 DumpPhilipsECHO.cxx	2359
14.55 DumpSiemensBase64.cxx	2364
14.56 DumpToSQLITE3.cxx	2365
14.57 DumpToshibaDTI.cxx	2367
14.58 DumpToshibaDTI2.cxx	2368
14.59 DumpVisusChange.cxx	2370
14.60 DuplicatePCDE.cxx	2372
14.61 ELSCINT1WaveToText.cxx	2374
14.62 EmptyMask.cxx	2376
14.63 EncapsulateFileInRawData.cxx	2377
14.64 ExtractEncryptedContent.cxx	2378
14.65 ExtractIconFromFile.cxx	2379
14.66 Extracting_All_Resolution.cxx	2380
14.67 Fake_Image_Using_Stream_Image_Writer.cxx	2386
14.68 FixBrokenJ2K.cxx	2389
14.69 FixJAIBugJPEGLS.cxx	2391
14.70 FixOrientation.cxx	2393
14.71 GenAllVR.cxx	2395
14.72 GenFakeIdentifyFile.cxx	2397
14.73 GenLongSeqs.cxx	2399
14.74 GenSeqs.cxx	2400
14.75 GenerateStandardSOPClasses.cxx	2402
14.76 GetJPEGSamplePrecision.cxx	2403
14.77 GetSequenceUltrasound.cxx	2404
14.78 GetSubSequenceData.cxx	2406
14.79 HelloVizWorld.cxx	2409
14.80 HelloWorld.cxx	2410
14.81 LargeVRDSExplicit.cxx	2411
14.82 MakeTemplate.cxx	2413
14.83 MergeTwoFiles.cxx	2414
14.84 MrProtocol.cxx	2415
14.85 PrintLUT.cxx	2422
14.86 PublicDict.cxx	2423
14.87 QIDO-RS.cxx	2424
14.88 ReadAndDumpDICOMDIR.cxx	2425
14.89 ReadAndDumpDICOMDIR2.cxx	2427
14.90 ReadAndPrintAttributes.cxx	2432
14.91 ReadExplicitLengthSQIVR.cxx	2434

14.92 ReadGEMSSDO.cxx	2434
14.93 ReadMultiTimesException.cxx	2437
14.94 ReadUTF8QtDir.cxx	2437
14.95 SimpleScanner.cxx	2439
14.96 SortImage.cxx	2441
14.97 StreamImageReaderTest.cxx	2442
14.98 TemplateEmptyImage.cxx	2446
14.99 TraverseModules.cxx	2447
14.100 VolumeSorter.cxx	2448
14.101 csa2img.cxx	2451
14.102 iU22tomultisc.cxx	2453
14.103 pmsct_rgb1.cxx	2454
14.104 rle2img.cxx	2458
14.105 uid_unique.cxx	2460
14.106 DecompressImage.java	2461
14.107 DecompressPixmap.java	2462
14.108 ExtractImageRegion.java	2463
14.109 FileAnonymize.java	2464
14.110 HelloSimple.java	2465
14.111 ReadFiles.java	2465
14.112 ScanDirectory.java	2467
14.113 SimplePrint.java	2470
14.114 AddPrivateAttribute.py	2471
14.115 ConvertMPL.py	2472
14.116 ConvertNumpy.py	2473
14.117 ConvertPIL.py	2474
14.118 CreateRAWStorage.py	2475
14.119 DecompressImage.py	2477
14.120 DumbAnonymizer.py	2478
14.121 ExtractImageRegion.py	2479
14.122 FindAllPatientName.py	2480
14.123 FixCommaBug.py	2481
14.124 GetPortionCSAHeader.py	2482
14.125 HelloWorld.py	2482
14.126 ManipulateFile.py	2483
14.127 ManipulateSequence.py	2484
14.128 MergeFile.py	2485
14.129 NewSequence.py	2486
14.130 PhilipsPrivateRescaleInterceptSlope.py	2487
14.131 PlaySound.py	2487

14.132 PrivateDict.py	2488
14.133 ReWriteSCAsMR.py	2489
14.134 ReadAndDumpDICOMDIR.py	2490
14.135 RemovePrivateTags.py	2492
14.136 ScanDirectory.py	2492
14.137 SortImage.py	2493
14.138 WriteBuffer.py	2494
14.139 HelloActiviz.cs	2495
14.140 HelloActiviz2.cs	2496
14.141 HelloActiviz3.cs	2497
14.142 HelloActiviz4.cs	2498
14.143 HelloActiviz5.cs	2499
14.144 HelloVTKWorld.cs	2500
14.145 HelloVTKWorld2.cs	2501
14.146 MetaImageMD5Activiz.cs	2501
14.147 RefCounting.cs	2503
14.148 Compute3DSpacing.cxx	2504
14.149 Convert16BitsTo8Bits.cxx	2505
14.150 ConvertMultiFrameToSingleFrame.cxx	2506
14.151 ConvertRGBToLuminance.cxx	2507
14.152 ConvertSingleBitTo8Bits.cxx	2508
14.153 CreateFakePET.cxx	2509
14.154 CreateFakeRTDOSE.cxx	2511
14.155 GenerateRTSTRUCT.cxx	2512
14.156 MagnifyFile.cxx	2515
14.157 gdcmmorthoplanes.cxx	2516
14.158 gdcmmreslice.cxx	2522
14.159 gdcmmrtionplan.cxx	2524
14.160 gdcmmrtplan.cxx	2528
14.161 gdcmmscene.cxx	2532
14.162 gdcmmtexture.cxx	2534
14.163 gdcmmvolume.cxx	2536
14.164 offscreenimage.cxx	2537
14.165 reslicesphere.cxx	2538
14.166 rtstructapp.cxx	2546
14.167 threadgdcmm.cxx	2548
14.168 AWTMedical3.java	2551
14.169 HelloVTKWorld.java	2555
14.170 MIPViewer.java	2557
14.171 MPRViewer.java	2559

14.172 MPRViewer2.java	2561
14.173 ReadSeriesIntoVTK.java	2565
14.174 CastConvertPhilips.py	2567
14.175 headsq2dcm.py	2569
Index	2571

Chapter 1

GDCM Documentation

This is the developers documentation.

A PDF version of this doxygen documentation can be found here:

<http://gdcml.sourceforge.net/3.2/gdcm-3.2.5.pdf>

A tarball version of this HTML doxygen documentation can be found here:

<http://gdcml.sourceforge.net/3.2/gdcm-3.2.5-doc.tar.gz>

Author

Mathieu Malaterre

Chapter 2

Todo List

Class [gdcm::CSAHeader](#)

MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

Class [gdcm::network::ApplicationContext](#)

Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

Class [gdcm::Overlay](#)

Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Class [gdcm::SequenceOfFragments](#)

I do not enforce that Sequence of Fragments ends with a SQ end del

Class [gdcm::TransferSyntax](#)

: The implementation is completely retarded -> see [gdcm::UIDs](#) for a replacement We need: IsSupported
We need preprocess of raw/xml file We need GetFullName()

Member [gdcm::UIDGenerator::IsValid](#) (const char *uid)

: Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

Chapter 3

Deprecated List

Member `gdcm::CompositeNetworkFunctions::ConstructQuery` (`ERootType` inRootType, `EQueryLevel` inQueryLevel, const `KeyValuePairArrayType` &keys, `EQueryType` queryType=eFind)

Member `gdcm::FileSet::AddFile` (`File` const &)

. Does nothing

Member `gdcm::TransferSyntax::GetSwapCode` () const

Return the `SwapCode` associated with the Transfer Syntax. Be careful with the special GE private syntax the `DataSet` is written in little endian but the Pixel Data is in Big Endian.

Chapter 4

Bug List

Class [gdcm::DICOMDIRGenerator](#)

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the [Scanner](#) does not allow us See PS 3.11 / [Table](#) D.3-2 STD-GEN Additional [DICOMDIR](#) Keys

Member [gdcm::FileStreamer::StartGroupDataElement](#) (const [PrivateTag](#) &pt, size_t maxsize=0, uint8_t startoffset=0)

maxsize should be a value lower than the actual total size of the buffer to be copied

Class [gdcm::IPPSorter](#)

There are currently a couple of bugs in this implementation:

Chapter 5

Directory Hierarchy

5.1 Directories

Common	57
gdcmlASN1.h	1575
gdcmlBase64.h	1577
gdcmlBoxRegion.h	1578
gdcmlByteSwap.h	1579
gdcmlCAPICryptoFactory.h	1581
gdcmlCAPICryptographicMessageSyntax.h	1582
gdcmlCommand.h	1585
gdcmlCryptoFactory.h	1588
gdcmlCryptographicMessageSyntax.h	1590
gdcmlDataEvent.h	1592
gdcmlDeflateStream.h	1594
gdcmlDirectory.h	1594
gdcmlDummyValueGenerator.h	1597
gdcmlEvent.h	1598
gdcmlException.h	1601
gdcmlFilename.h	1605
gdcmlFileNameEvent.h	1606
gdcmlFilenameGenerator.h	1608
gdcmlLegacyMacro.h	1609
gdcmlMD5.h	1612
gdcmlObject.h	1613
gdcmlOpenSSLCryptoFactory.h	1616
gdcmlOpenSSLCryptographicMessageSyntax.h	1617
gdcmlOpenSSL7CryptoFactory.h	1619
gdcmlOpenSSL7CryptographicMessageSyntax.h	1621
gdcmlProgressEvent.h	1623
gdcmlRegion.h	1624
gdcmlSHA1.h	1627
gdcmlSmartPointer.h	1628
gdcmlStaticAssert.h	1630
gdcmlString.h	1632
gdcmlSubject.h	1636
gdcmlSwapCode.h	1637

gdcmswapper.h	1639
gdcmsystem.h	1642
gdcmterminal.h	1644
gdcmtestdriver.h	1646
gdcmtesting.h	1647
gdcmtrace.h	1649
gdcmtypes.h	1655
gdcmunpacker12bits.h	1656
gdcmversion.h	1657
gdcmwin32.h	1659
DataDictionary	59
gdcmsaheaderdict.h	1660
gdcmsaheaderdictentry.h	1663
gdcmdict.h	1666
gdcmdictconverter.h	1671
gdcmdictentry.h	1673
gdcmdicts.h	1676
gdcmglobal.h	1679
gdcmgrouptdict.h	1681
gdcmsopclassuidtoiod.h	1683
gdcmuuids.h	1684
DataStructureAndEncodingDefinition	60
gdcmattribute.h	1697
gdcmbasicoffsettable.h	1712
gdcmbytebuffer.h	1714
gdcmbyteswapfilter.h	1717
gdcmbytevalue.h	1718
gdcmcodestring.h	1723
gdcmp246explicitdataelement.h	1725
gdcmsaelement.h	1726
gdcmsaheader.h	1729
gdcmdataelement.h	1732
gdcmdataset.h	1735
gdcmdatasetevent.h	1740
gdcmelement.h	1742
gdcmexplicitdataelement.h	1754
gdcmexplicitimplicitdataelement.h	1756
gdcmfile.h	1758
gdcmfilemetainformation.h	1759
gdcmfileset.h	1762
gdcmfragment.h	1764
gdcmimplicitdataelement.h	1769
gdcmitem.h	1770
gdcml.o.h	1776
gdcmmediastorage.h	1777
gdcmmrprotocol.h	1781
gdcmparseexception.h	1783
gdcmparser.h	1785
gdcmpdbelement.h	1788
gdcmpdbheader.h	1790
gdcmpreamble.h	1792
gdcmprivatetag.h	1794
gdcmreader.h	1796
gdcmsequenceoffragments.h	1798

gdcSequenceOfItems.h	1803
gdcTag.h	1807
gdcTagToVR.h	1812
gdcTransferSyntax.h	1813
gdcUNExplicitDataElement.h	1815
gdcUNExplicitImplicitDataElement.h	1817
gdcValue.h	1818
gdcValueIO.h	1820
gdcVL.h	1821
gdcVM.h	1824
gdcVR.h	1827
gdcVR16ExplicitDataElement.h	1834
gdcWriter.h	1836
InformationObjectDefinition	61
gdcDefinedTerms.h	1838
gdcDefs.h	1839
gdcEnumeratedValues.h	1842
gdcIOD.h	1843
gdcIODEntry.h	1846
gdcIODs.h	1848
gdcMacro.h	1851
gdcMacroEntry.h	1854
gdcMacros.h	1857
gdcModule.h	1859
gdcModuleEntry.h	1862
gdcModules.h	1865
gdcNestedModuleEntries.h	1867
gdcPatient.h	1869
gdcSeries.h	1871
gdcStudy.h	1873
gdcTable.h	1874
gdcTableEntry.h	1876
gdcTableReader.h	1878
gdcType.h	1881
gdcUsage.h	1883
gdcXMLDictReader.h	1886
gdcXMLPrivateDictReader.h	1888
MediaStorageAndFileFormat	62
gdcAnonymizeEvent.h	1889
gdcAnonymizer.h	1891
gdcApplicationEntity.h	1894
gdcAudioCodec.h	1895
gdcBitmap.h	1896
gdcBitmapToBitmapFilter.h	1900
gdcCleaner.h	1901
gdcCodec.h	1903
gdcCoder.h	1904
gdcConstCharWrapper.h	1906
gdcCurve.h	1907
gdcDataSetHelper.h	1910
gdcDecoder.h	1911
gdcDeltaEncodingCodec.h	1913
gdcDICOMDIR.h	1914
gdcDICOMDIRGenerator.h	1915

gdcmDictPrinter.h	1917
gdcmDirectionCosines.h	1918
gdcmDirectoryHelper.h	1920
gdcmDPath.h	1921
gdcmDumper.h	1924
gdcmEmptyMaskGenerator.h	1925
gdcmEncapsulatedDocument.h	1927
gdcmEquipmentManufacturer.h	1928
gdcmFiducials.h	1929
gdcmFileAnonymizer.h	1930
gdcmFileChangeTransferSyntax.h	1932
gdcmFileDecompressLookupTable.h	1934
gdcmFileDerivation.h	1936
gdcmFileExplicitFilter.h	1938
gdcmFileStreamer.h	1939
gdcmIconImage.h	1941
gdcmIconImageFilter.h	1943
gdcmIconImageGenerator.h	1945
gdcmImage.h	1946
gdcmImageApplyLookupTable.h	1949
gdcmImageChangePhotometricInterpretation.h	1950
gdcmImageChangePlanarConfiguration.h	1953
gdcmImageChangeTransferSyntax.h	1954
gdcmImageCodec.h	1956
gdcmImageConverter.h	1959
gdcmImageFragmentSplitter.h	1961
gdcmImageHelper.h	1962
gdcmImageReader.h	1964
gdcmImageRegionReader.h	1966
gdcmImageToImageFilter.h	1968
gdcmImageWriter.h	1969
gdcmIPPSorter.h	1971
gdcmJPEG12Codec.h	1973
gdcmJPEG16Codec.h	1974
gdcmJPEG2000Codec.h	1976
gdcmJPEG8Codec.h	1978
gdcmJPEGCodec.h	1979
gdcmJPEGLSCodec.h	1982
gdcmJSON.h	1983
gdcmKAKADUCodec.h	1985
gdcmLookupTable.h	1986
gdcmMEC_MR3.h	1989
gdcmMeshPrimitive.h	1990
gdcmOrientation.h	1993
gdcmOverlay.h	1994
gdcmPDFCodec.h	1997
gdcmPersonName.h	1998
gdcmPGXCodec.h	2000
gdcmPhotometricInterpretation.h	2001
gdcmPixelFormat.h	2004
gdcmPixmap.h	2008
gdcmPixmapReader.h	2010
gdcmPixmapToPixmapFilter.h	2012
gdcmPixmapWriter.h	2013

gdcmPNMCodec.h	2016
gdcmPrinter.h	2017
gdcmPVRGCodec.h	2020
gdcmRAWCodec.h	2021
gdcmRescaler.h	2022
gdcmRLECodec.h	2025
gdcmScanner.h	2026
gdcmScanner2.h	2029
gdcmSegment.h	2032
gdcmSegmentedPaletteColorLookupTable.h	2036
gdcmSegmentHelper.h	2037
gdcmSegmentReader.h	2039
gdcmSegmentWriter.h	2041
gdcmSerieHelper.h	2043
gdcmSimpleSubjectWatcher.h	2046
gdcmSorter.h	2048
gdcmSpacing.h	2051
gdcmSpectroscopy.h	2053
gdcmSplitMosaicFilter.h	2054
gdcmStreamImageReader.h	2056
gdcmStreamImageWriter.h	2057
gdcmStrictScanner.h	2059
gdcmStrictScanner2.h	2062
gdcmStringFilter.h	2065
gdcmSurface.h	2066
gdcmSurfaceHelper.h	2071
gdcmSurfaceReader.h	2073
gdcmSurfaceWriter.h	2075
gdcmTagPath.h	2076
gdcmUIDGenerator.h	2078
gdcmUUIDGenerator.h	2080
gdcmValidate.h	2081
gdcmWaveform.h	2082
gdcmXMLPrinter.h	2083
MessageExchangeDefinition	65
gdcmAAAbortPDU.h	2086
gdcmAAAssociateACPDU.h	2087
gdcmAAAssociateRJPDU.h	2090
gdcmAAAssociateRQPDU.h	2091
gdcmAbstractSyntax.h	2094
gdcmApplicationContext.h	2096
gdcmAReleaseRPPDU.h	2097
gdcmAReleaseRQPDU.h	2099
gdcmARTIMTimer.h	2101
gdcmAsynchronousOperationsWindowSub.h	2102
gdcmBaseCompositeMessage.h	2103
gdcmBaseNormalizedMessage.h	2105
gdcmBasePDU.h	2107
gdcmBaseQuery.h	2108
gdcmBaseRootQuery.h	2111
gdcmCEchoMessages.h	2113
gdcmCFindMessages.h	2114
gdcmCMoveMessages.h	2116
gdcmCommandDataSet.h	2118

gdcmCompositeMessageFactory.h	2119
gdcmCompositeNetworkFunctions.h	2121
gdcmCStoreMessages.h	2123
gdcmDIMSE.h	2124
gdcmFindPatientRootQuery.h	2127
gdcmFindStudyRootQuery.h	2128
gdcmImplementationClassUIDSub.h	2130
gdcmImplementationUIDSub.h	2132
gdcmImplementationVersionNameSub.h	2133
gdcmMaximumLengthSub.h	2135
gdcmModalityPerformedProcedureStepCreateQuery.h	2137
gdcmModalityPerformedProcedureStepSetQuery.h	2138
gdcmMovePatientRootQuery.h	2140
gdcmMoveStudyRootQuery.h	2141
gdcmNActionMessages.h	2142
gdcmNCreateMessages.h	2144
gdcmNDeleteMessages.h	2145
gdcmNetworkEvents.h	2146
gdcmNetworkStateID.h	2148
gdcmNEventReportMessages.h	2150
gdcmNGetMessages.h	2152
gdcmNormalizedMessageFactory.h	2153
gdcmNormalizedNetworkFunctions.h	2154
gdcmNSetMessages.h	2156
gdcmPDataTFPDU.h	2157
gdcmPDUFactory.h	2159
gdcmPresentationContext.h	2161
gdcmPresentationContextAC.h	2163
gdcmPresentationContextGenerator.h	2165
gdcmPresentationContextRQ.h	2166
gdcmPresentationDataValue.h	2169
gdcmQueryBase.h	2171
gdcmQueryFactory.h	2173
gdcmQueryImage.h	2175
gdcmQueryPatient.h	2177
gdcmQuerySeries.h	2179
gdcmQueryStudy.h	2181
gdcmRoleSelectionSub.h	2182
gdcmServiceClassApplicationInformation.h	2184
gdcmServiceClassUser.h	2185
gdcmSOPClassExtendedNegociationSub.h	2188
gdcmTransferSyntaxSub.h	2189
gdcmULAction.h	2191
gdcmULActionAA.h	2193
gdcmULActionAE.h	2195
gdcmULActionAR.h	2197
gdcmULActionDT.h	2200
gdcmULBasicCallback.h	2201
gdcmULConnection.h	2202
gdcmULConnectionCallback.h	2205
gdcmULConnectionInfo.h	2206
gdcmULConnectionManager.h	2208
gdcmULEvent.h	2211
gdcmULTransitionTable.h	2213

gdcmlULWritingCallback.h	2216
gdcmlUserInformation.h	2217
gdcmlWLMFindQuery.h	2219
Python	67
gdcmlPythonFilter.h	2266
Source	67
Common	57
gdcmlASN1.h	1575
gdcmlBase64.h	1577
gdcmlBoxRegion.h	1578
gdcmlByteSwap.h	1579
gdcmlCAPICryptoFactory.h	1581
gdcmlCAPICryptographicMessageSyntax.h	1582
gdcmlCommand.h	1585
gdcmlCryptoFactory.h	1588
gdcmlCryptographicMessageSyntax.h	1590
gdcmlDataEvent.h	1592
gdcmlDeflateStream.h	1594
gdcmlDirectory.h	1594
gdcmlDummyValueGenerator.h	1597
gdcmlEvent.h	1598
gdcmlException.h	1601
gdcmlFilename.h	1605
gdcmlFileNameEvent.h	1606
gdcmlFilenameGenerator.h	1608
gdcmlLegacyMacro.h	1609
gdcmlMD5.h	1612
gdcmlObject.h	1613
gdcmlOpenSSLCryptoFactory.h	1616
gdcmlOpenSSLCryptographicMessageSyntax.h	1617
gdcmlOpenSSL7CryptoFactory.h	1619
gdcmlOpenSSL7CryptographicMessageSyntax.h	1621
gdcmlProgressEvent.h	1623
gdcmlRegion.h	1624
gdcmlSHA1.h	1627
gdcmlSmartPointer.h	1628
gdcmlStaticAssert.h	1630
gdcmlString.h	1632
gdcmlSubject.h	1636
gdcmlSwapCode.h	1637
gdcmlSwapper.h	1639
gdcmlSystem.h	1642
gdcmlTerminal.h	1644
gdcmlTestDriver.h	1646
gdcmlTesting.h	1647
gdcmlTrace.h	1649
gdcmlTypes.h	1655
gdcmlUnpacker12Bits.h	1656
gdcmlVersion.h	1657
gdcmlWin32.h	1659
DataDictionary	59
gdcmlCSAHeaderDict.h	1660
gdcmlCSAHeaderDictEntry.h	1663

gdcmDict.h	1666
gdcmDictConverter.h	1671
gdcmDictEntry.h	1673
gdcmDicts.h	1676
gdcmGlobal.h	1679
gdcmGroupDict.h	1681
gdcmSOPClassUIDToIOD.h	1683
gdcmUIDs.h	1684
DataStructureAndEncodingDefinition	60
gdcmAttribute.h	1697
gdcmBasicOffsetTable.h	1712
gdcmByteBuffer.h	1714
gdcmByteSwapFilter.h	1717
gdcmByteValue.h	1718
gdcmCodeString.h	1723
gdcmCP246ExplicitDataElement.h	1725
gdcmCSAElement.h	1726
gdcmCSAHeader.h	1729
gdcmDataElement.h	1732
gdcmDataSet.h	1735
gdcmDataSetEvent.h	1740
gdcmElement.h	1742
gdcmExplicitDataElement.h	1754
gdcmExplicitImplicitDataElement.h	1756
gdcmFile.h	1758
gdcmFileMetaInformation.h	1759
gdcmFileSet.h	1762
gdcmFragment.h	1764
gdcmImplicitDataElement.h	1769
gdcmItem.h	1770
gdcmLO.h	1776
gdcmMediaStorage.h	1777
gdcmMrProtocol.h	1781
gdcmParseException.h	1783
gdcmParser.h	1785
gdcmPDBelement.h	1788
gdcmPDBHeader.h	1790
gdcmPreamble.h	1792
gdcmPrivateTag.h	1794
gdcmReader.h	1796
gdcmSequenceOfFragments.h	1798
gdcmSequenceOfItems.h	1803
gdcmTag.h	1807
gdcmTagToVR.h	1812
gdcmTransferSyntax.h	1813
gdcmUNExplicitDataElement.h	1815
gdcmUNExplicitImplicitDataElement.h	1817
gdcmValue.h	1818
gdcmValueIO.h	1820
gdcmVL.h	1821
gdcmVM.h	1824
gdcmVR.h	1827
gdcmVR16ExplicitDataElement.h	1834
gdcmWriter.h	1836

InformationObjectDefinition	61
gdcmlDefinedTerms.h	1838
gdcmlDefs.h	1839
gdcmlEnumeratedValues.h	1842
gdcmlIOD.h	1843
gdcmlIODEntry.h	1846
gdcmlIODs.h	1848
gdcmlMacro.h	1851
gdcmlMacroEntry.h	1854
gdcmlMacros.h	1857
gdcmlModule.h	1859
gdcmlModuleEntry.h	1862
gdcmlModules.h	1865
gdcmlNestedModuleEntries.h	1867
gdcmlPatient.h	1869
gdcmlSeries.h	1871
gdcmlStudy.h	1873
gdcmlTable.h	1874
gdcmlTableEntry.h	1876
gdcmlTableReader.h	1878
gdcmlType.h	1881
gdcmlUsage.h	1883
gdcmlXMLDictReader.h	1886
gdcmlXMLPrivateDictReader.h	1888
MediaStorageAndFileFormat	62
gdcmlAnonymizeEvent.h	1889
gdcmlAnonymizer.h	1891
gdcmlApplicationEntity.h	1894
gdcmlAudioCodec.h	1895
gdcmlBitmap.h	1896
gdcmlBitmapToBitmapFilter.h	1900
gdcmlCleaner.h	1901
gdcmlCodec.h	1903
gdcmlCoder.h	1904
gdcmlConstCharWrapper.h	1906
gdcmlCurve.h	1907
gdcmlDataSetHelper.h	1910
gdcmlDecoder.h	1911
gdcmlDeltaEncodingCodec.h	1913
gdcmlDICOmdir.h	1914
gdcmlDICOmdirGenerator.h	1915
gdcmlDictPrinter.h	1917
gdcmlDirectionCosines.h	1918
gdcmlDirectoryHelper.h	1920
gdcmlDPath.h	1921
gdcmlDumper.h	1924
gdcmlEmptyMaskGenerator.h	1925
gdcmlEncapsulatedDocument.h	1927
gdcmlEquipmentManufacturer.h	1928
gdcmlFiducials.h	1929
gdcmlFileAnonymizer.h	1930
gdcmlFileChangeTransferSyntax.h	1932
gdcmlFileDecompressLookupTable.h	1934
gdcmlFileDerivation.h	1936

gdcmFileExplicitFilter.h	1938
gdcmFileStreamer.h	1939
gdcmIconImage.h	1941
gdcmIconImageFilter.h	1943
gdcmIconImageGenerator.h	1945
gdcmImage.h	1946
gdcmImageApplyLookupTable.h	1949
gdcmImageChangePhotometricInterpretation.h	1950
gdcmImageChangePlanarConfiguration.h	1953
gdcmImageChangeTransferSyntax.h	1954
gdcmImageCodec.h	1956
gdcmImageConverter.h	1959
gdcmImageFragmentSplitter.h	1961
gdcmImageHelper.h	1962
gdcmImageReader.h	1964
gdcmImageRegionReader.h	1966
gdcmImageToImageFilter.h	1968
gdcmImageWriter.h	1969
gdcmIPPSorter.h	1971
gdcm.JPEG12Codec.h	1973
gdcm.JPEG16Codec.h	1974
gdcm.JPEG2000Codec.h	1976
gdcm.JPEG8Codec.h	1978
gdcm.JPEGCodec.h	1979
gdcm.JPEGLSCodec.h	1982
gdcm.JSON.h	1983
gdcmKAKADUCodec.h	1985
gdcmLookupTable.h	1986
gdcmMEC_MR3.h	1989
gdcmMeshPrimitive.h	1990
gdcmOrientation.h	1993
gdcmOverlay.h	1994
gdcmPDFCodec.h	1997
gdcmPersonName.h	1998
gdcmPGXCodec.h	2000
gdcmPhotometricInterpretation.h	2001
gdcmPixelFormat.h	2004
gdcmPixmap.h	2008
gdcmPixmapReader.h	2010
gdcmPixmapToPixmapFilter.h	2012
gdcmPixmapWriter.h	2013
gdcmPNMCodec.h	2016
gdcmPrinter.h	2017
gdcmPVRGCodec.h	2020
gdcmRAWCodec.h	2021
gdcmRescaler.h	2022
gdcmRLECodec.h	2025
gdcmScanner.h	2026
gdcmScanner2.h	2029
gdcmSegment.h	2032
gdcmSegmentedPaletteColorLookupTable.h	2036
gdcmSegmentHelper.h	2037
gdcmSegmentReader.h	2039
gdcmSegmentWriter.h	2041

gdcmSerieHelper.h	2043
gdcmSimpleSubjectWatcher.h	2046
gdcmSorter.h	2048
gdcmSpacing.h	2051
gdcmSpectroscopy.h	2053
gdcmSplitMosaicFilter.h	2054
gdcmStreamImageReader.h	2056
gdcmStreamImageWriter.h	2057
gdcmStrictScanner.h	2059
gdcmStrictScanner2.h	2062
gdcmStringFilter.h	2065
gdcmSurface.h	2066
gdcmSurfaceHelper.h	2071
gdcmSurfaceReader.h	2073
gdcmSurfaceWriter.h	2075
gdcmTagPath.h	2076
gdcmUIDGenerator.h	2078
gdcmUUIDGenerator.h	2080
gdcmValidate.h	2081
gdcmWaveform.h	2082
gdcmXMLPrinter.h	2083
MessageExchangeDefinition	65
gdcmAAabortPDU.h	2086
gdcmAAssociateACPDU.h	2087
gdcmAAssociateRJPDU.h	2090
gdcmAAssociateRQPDU.h	2091
gdcmAbstractSyntax.h	2094
gdcmApplicationContext.h	2096
gdcmAReleaseRPPDU.h	2097
gdcmAReleaseRQPDU.h	2099
gdcmARTIMTimer.h	2101
gdcmAsynchronousOperationsWindowSub.h	2102
gdcmBaseCompositeMessage.h	2103
gdcmBaseNormalizedMessage.h	2105
gdcmBasePDU.h	2107
gdcmBaseQuery.h	2108
gdcmBaseRootQuery.h	2111
gdcmCEchoMessages.h	2113
gdcmCFindMessages.h	2114
gdcmCMoveMessages.h	2116
gdcmCommandDataSet.h	2118
gdcmCompositeMessageFactory.h	2119
gdcmCompositeNetworkFunctions.h	2121
gdcmCStoreMessages.h	2123
gdcmDIMSE.h	2124
gdcmFindPatientRootQuery.h	2127
gdcmFindStudyRootQuery.h	2128
gdcmImplementationClassUIDSub.h	2130
gdcmImplementationUIDSub.h	2132
gdcmImplementationVersionNameSub.h	2133
gdcmMaximumLengthSub.h	2135
gdcmModalityPerformedProcedureStepCreateQuery.h	2137
gdcmModalityPerformedProcedureStepSetQuery.h	2138
gdcmMovePatientRootQuery.h	2140

gdcmoveStudyRootQuery.h	2141
gdcnActionMessages.h	2142
gdcnCreateMessages.h	2144
gdcnDeleteMessages.h	2145
gdcnetworkEvents.h	2146
gdcnetworkStateID.h	2148
gdcnEventReportMessages.h	2150
gdcnGetMessages.h	2152
gdcnormalizedMessageFactory.h	2153
gdcnormalizedNetworkFunctions.h	2154
gdcnSetMessages.h	2156
gdcpDataTFPDU.h	2157
gdcpDUFactory.h	2159
gdcpresentationContext.h	2161
gdcpresentationContextAC.h	2163
gdcpresentationContextGenerator.h	2165
gdcpresentationContextRQ.h	2166
gdcpresentationDataValue.h	2169
gdcqueryBase.h	2171
gdcqueryFactory.h	2173
gdcqueryImage.h	2175
gdcqueryPatient.h	2177
gdcquerySeries.h	2179
gdcqueryStudy.h	2181
gdcroleSelectionSub.h	2182
gdcserviceClassApplicationInformation.h	2184
gdcserviceClassUser.h	2185
gdcsopClassExtendedNegociationSub.h	2188
gdctransferSyntaxSub.h	2189
gdcULAction.h	2191
gdcULActionAA.h	2193
gdcULActionAE.h	2195
gdcULActionAR.h	2197
gdcULActionDT.h	2200
gdcULBasicCallback.h	2201
gdcULConnection.h	2202
gdcULConnectionCallback.h	2205
gdcULConnectionInfo.h	2206
gdcULConnectionManager.h	2208
gdcULEvent.h	2211
gdcULTransitionTable.h	2213
gdcULWritingCallback.h	2216
gdcuserInformation.h	2217
gdcmWLMFindQuery.h	2219
Utilities	68
VTK	68
vtkGDCMImageReader.h	2220
vtkGDCMImageReader2.h	2226
vtkGDCMImageWriter.h	2231
vtkGDCMMedicalImageProperties.h	2234
vtkGDCMPolyDataReader.h	2240
vtkGDCMPolyDataWriter.h	2241
vtkGDCMTesting.h	2243
vtkGDCMThreadedImageReader.h	2245

vtkGDCMThreadedImageReader2.h	2247
vtkImageColorViewer.h	2249
vtkImageMapToColors16.h	2253
vtkImageMapToWindowLevelColors2.h	2255
vtkImagePlanarComponentsToComponents.h	2257
vtkImageRGBToYBR.h	2259
vtkImageYBRToRGB.h	2261
vtkLookupTable16.h	2262
vtkRTStructSetProperties.h	2264
VTK	68
vtkGDCMImageReader.h	2220
vtkGDCMImageReader2.h	2226
vtkGDCMImageWriter.h	2231
vtkGDCMMedicalImageProperties.h	2234
vtkGDCMPolyDataReader.h	2240
vtkGDCMPolyDataWriter.h	2241
vtkGDCMTesting.h	2243
vtkGDCMThreadedImageReader.h	2245
vtkGDCMThreadedImageReader2.h	2247
vtkImageColorViewer.h	2249
vtkImageMapToColors16.h	2253
vtkImageMapToWindowLevelColors2.h	2255
vtkImagePlanarComponentsToComponents.h	2257
vtkImageRGBToYBR.h	2259
vtkImageYBRToRGB.h	2261
vtkLookupTable16.h	2262
vtkRTStructSetProperties.h	2264
Wrapping	69
Python	67
gdcmlPythonFilter.h	2266

Chapter 6

Namespace Index

6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdc	71
gdc::network	102
gdc::SegmentHelper	108
gdc::terminal	
Class for Terminal	109

Chapter 7

Hierarchical Index

7.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

gdcmm::network::AbstractSyntax	129
gdcmm::network::ApplicationContext	145
gdcmm::ApplicationEntity	147
gdcmm::network::ARTIMTimer	154
gdcmm::ASN1	155
gdcmm::network::AsynchronousOperationsWindowSub	157
gdcmm::Attribute< Group, Element, TVR, TVM >	158
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >	168
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >	187
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >	176
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >	181
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >	201
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >	194
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >	214
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >	207
gdcmm::Base64	222
gdcmm::network::BaseCompositeMessage	224
gdcmm::network::CEchoRQ	289
gdcmm::network::CEchoRSP	291
gdcmm::network::CFindCancelRQ	293
gdcmm::network::CFindRQ	294
gdcmm::network::CFindRSP	296
gdcmm::network::CMoveCancelRq	304
gdcmm::network::CMoveRQ	306
gdcmm::network::CMoveRSP	307
gdcmm::network::CStoreRQ	361
gdcmm::network::CStoreRSP	363
gdcmm::network::BaseNormalizedMessage	226
gdcmm::network::NActionRQ	846
gdcmm::network::NActionRSP	847

gdcmm::network::NCreateRQ	849
gdcmm::network::NCreateRSP	850
gdcmm::network::NDeleteRQ	852
gdcmm::network::NDeleteRSP	853
gdcmm::network::NEventReportRQ	858
gdcmm::network::NEventReportRSP	859
gdcmm::network::NGetRQ	861
gdcmm::network::NGetRSP	862
gdcmm::network::NSetRQ	869
gdcmm::network::NSetRSP	870
gdcmm::network::BasePDU	229
gdcmm::network::AAAbortPDU	113
gdcmm::network::AAAssociateACPDU	116
gdcmm::network::AAAssociateRJPDU	120
gdcmm::network::AAAssociateRQPDU	122
gdcmm::network::AReleaseRPPDU	149
gdcmm::network::AReleaseRQPDU	151
gdcmm::network::PDataTFPDU	905
std::basic_string< Char >	
std::string	
gdcmm::String< '\', 16 >	1189
gdcmm::String< '\', 64 >	1189
gdcmm::String< '\', 4294967294 >	1189
gdcmm::String< '\', 64, 0 >	1189
gdcmm::String< TDelimiter, TMaxLength, TPadChar >	1189
gdcmm::SegmentHelper::BasicCodedEntry	241
gdcmm::BitmapToBitmapFilter	263
gdcmm::PixmapToPixmapFilter	950
gdcmm::ImageToImageFilter	690
gdcmm::ImageApplyLookupTable	637
gdcmm::ImageChangePhotometricInterpretation	640
gdcmm::ImageChangePlanarConfiguration	645
gdcmm::ImageChangeTransferSyntax	649
gdcmm::ImageFragmentSplitter	670
gdcmm::ByteBuffer	270
gdcmm::ByteSwap< T >	271
gdcmm::ByteSwapFilter	273
gdcmm::network::CFind	292
gdcmm::Coder	310
gdcmm::Codec	309
gdcmm::AudioCodec	219
gdcmm::ImageCodec	656
gdcmm::DeltaEncodingCodec	411
gdcmm::JPEG2000Codec	738
gdcmm::JPEGCodec	751
gdcmm::JPEG12Codec	727
gdcmm::JPEG16Codec	733
gdcmm::JPEG8Codec	746
gdcmm::JPEGLSCCodec	760
gdcmm::KAKADUCodec	769
gdcmm::PGXCodec	922
gdcmm::PNMCodec	958
gdcmm::PVRGCodec	999

gdcmm::RAWCodec	1019
gdcmm::RLECodec	1040
gdcmm::PDFCodec	914
gdcmm::CodeString	312
gdcmm::network::CompositeMessageFactory	324
gdcmm::CompositeNetworkFunctions	325
gdcmm::ConstCharWrapper	330
gdcmm::CryptoFactory	335
gdcmm::CAPICryptoFactory	284
gdcmm::OpenSSLCryptoFactory	875
gdcmm::OpenSSLP7CryptoFactory	881
gdcmm::CryptographicMessageSyntax	337
gdcmm::CAPICryptographicMessageSyntax	286
gdcmm::OpenSSLCryptographicMessageSyntax	877
gdcmm::OpenSSLP7CryptographicMessageSyntax	883
gdcmm::CSAElement	341
gdcmm::CSAHeader	348
gdcmm::CSAHeaderDict	354
gdcmm::CSAHeaderDictEntry	357
gdcmm::DataElement	369
gdcmm::CP246ExplicitDataElement	331
gdcmm::ExplicitDataElement	539
gdcmm::ExplicitImplicitDataElement	543
gdcmm::Fragment	610
gdcmm::BasicOffsetTable	244
gdcmm::ImplicitDataElement	701
gdcmm::Item	720
gdcmm::UNExplicitDataElement	1395
gdcmm::UNExplicitImplicitDataElement	1399
gdcmm::VR16ExplicitDataElement	1437
gdcmm::DataSet	388
gdcmm::CommandDataSet	320
gdcmm::FileMetaInformation	571
gdcmm::DataSetHelper	404
gdcmm::Decoder	405
gdcmm::Codec	309
gdcmm::DefinedTerms	406
gdcmm::Defs	407
gdcmm::DICOMDIR	415
gdcmm::DICOMDIRGenerator	416
gdcmm::Dict	419
gdcmm::DictConverter	424
gdcmm::DictEntry	428
gdcmm::Dicts	435
gdcmm::network::DIMSE	439
gdcmm::DirectionCosines	440
gdcmm::Directory	444
gdcmm::DirectoryHelper	448
gdcmm::DPath	450
gdcmm::DummyValueGenerator	452
gdcmm::Element< TVR, TVM >	456
gdcmm::Element< TVR, VM::VM1_n >	467

gdcmm::Element< TVR, VM::VM1_2 >	462
gdcmm::Element< TVR, VM::VM2_n >	480
gdcmm::Element< TVR, VM::VM2_2n >	474
gdcmm::Element< TVR, VM::VM3_4 >	492
gdcmm::Element< TVR, VM::VM3_n >	497
gdcmm::Element< TVR, VM::VM3_3n >	485
gdcmm::Element< VR::AS, VM::VM5 >	503
gdcmm::Element< VR::OB, VM::VM1_n >	456
gdcmm::Element< VR::OB, VM::VM1 >	507
gdcmm::Element< VR::OW, VM::VM1_n >	456
gdcmm::Element< VR::OW, VM::VM1 >	512
gdcmm::ElementDisableCombinations< TVR, TVM >	517
gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >	518
gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >	519
gdcmm::EmptyMaskGenerator	520
gdcmm::EncapsulatedDocument	522
gdcmm::EncodingImplementation< T >	523
gdcmm::EncodingImplementation< VR::VRASCII >	524
gdcmm::EncodingImplementation< VR::VRBINARY >	526
gdcmm::EnumeratedValues	529
gdcmm::EquipmentManufacturer	530
gdcmm::Event	532
gdcmm::AnyEvent	144
gdcmm::AbortEvent	128
gdcmm::AnonymizeEvent	131
gdcmm::DataEvent	384
gdcmm::DataSetEvent	400
gdcmm::EndEvent	528
gdcmm::ExitEvent	538
gdcmm::FileNameEvent	584
gdcmm::InitializeEvent	705
gdcmm::IterationEvent	726
gdcmm::ModifiedEvent	825
gdcmm::ProgressEvent	995
gdcmm::StartEvent	1155
gdcmm::UserEvent	1407
gdcmm::NoEvent	864
std::exception	
gdcmm::CSAHeaderDictException	360
gdcmm::DataElementException	384
gdcmm::Exception	535
gdcmm::ParseException	898
gdcmm::Fiducials	547
gdcmm::FileDerivation	564
gdcmm::FileExplicitFilter	568
gdcmm::Filename	581
gdcmm::FilenameGenerator	588
gdcmm::FileSet	591
gdcmm::Global	615
gdcmm::GroupDict	618
gdcmm::IconImageFilter	621
gdcmm::IconImageGenerator	624
gdcmm::ignore_char	627

gdcmm::ImageConverter	668
gdcmm::ImageHelper	673
gdcmm::network::ImplementationClassUIDSub	697
gdcmm::network::ImplementationUIDSub	699
gdcmm::network::ImplementationVersionNameSub	699
gdcmm::IOD	706
gdcmm::IODEntry	709
gdcmm::IODs	711
gdcmm::JSON	767
gdcmm::Scanner2::ltstr	784
gdcmm::Scanner::ltstr	785
gdcmm::StrictScanner2::ltstr	785
gdcmm::StrictScanner::ltstr	786
gdcmm::Macro	787
gdcmm::Macros	789
gdcmm::network::MaximumLengthSub	791
gdcmm::MD5	793
gdcmm::MEC_MR3	794
gdcmm::MediaStorage	795
gdcmm::Module	826
gdcmm::ModuleEntry	829
gdcmm::NestedModuleEntries	855
gdcmm::Modules	833
gdcmm::MrProtocol	844
gdcmm::network::NormalizedMessageFactory	865
gdcmm::NormalizedNetworkFunctions	866
gdcmm::Object	872
gdcmm::BaseQuery	231
gdcmm::BaseRootQuery	236
gdcmm::FindPatientRootQuery	602
gdcmm::FindStudyRootQuery	606
gdcmm::MovePatientRootQuery	836
gdcmm::MoveStudyRootQuery	840
gdcmm::WLMFindQuery	1554
gdcmm::ModalityPerformedProcedureStepCreateQuery	817
gdcmm::ModalityPerformedProcedureStepSetQuery	821
gdcmm::Bitmap	248
gdcmm::Pixmap	939
gdcmm::Image	628
gdcmm::Curve	364
gdcmm::File	548
gdcmm::FileWithName	599
gdcmm::LookupTable	777
gdcmm::SegmentedPaletteColorLookupTable	1078
gdcmm::MeshPrimitive	812
gdcmm::Overlay	889
gdcmm::Segment	1069
gdcmm::Subject	1198
gdcmm::Anonymizer	135
gdcmm::Cleaner	297
gdcmm::Command	316
gdcmm::MemberCommand< SimpleSubjectWatcher >	805
gdcmm::SimpleMemberCommand< SimpleSubjectWatcher >	1126

gdcmmembercommand< T >	805
gdcmmembercommand< T >	1126
gdcmmembercommand	553
gdcmmembercommand	557
gdcmmembercommand	561
gdcmmembercommand	593
gdcmmembercommand	1049
gdcmmembercommand	1058
gdcmmembercommand	1116
gdcmmembercommand	1169
gdcmmembercommand	1178
gdcmmembercommand	1382
gdcmmembercommand	1201
gdcmmembercommand	1414
gdcmmembercommand	275
gdcmmembercommand	1092
gdcmmembercommand	1099
gdcmmembercommand	886
gdcmmembercommand	900
gdcmmembercommand	904
gdcmmembercommand	908
gdcmmembercommand	911
gdcmmembercommand	916
gdcmmembercommand	920
gdcmmembercommand	926
gdcmmembercommand	930
gdcmmembercommand	963
gdcmmembercommand	967
gdcmmembercommand	970
gdcmmembercommand	972
gdcmmembercommand	975
gdcmmembercommand	979
gdcmmembercommand	983
gdcmmembercommand	432
gdcmmembercommand	453
gdcmmembercommand	987
gdcmmembercommand	1003
gdcmmembercommand	1005
gdcmmembercommand	1009
gdcmmembercommand	1012
gdcmmembercommand	1014
gdcmmembercommand	1017
gdcmmembercommand	1008
gdcmmembercommand	1024
gdcmmembercommand	946
gdcmmembercommand	680
gdcmmembercommand	685
gdcmmembercommand	1082
gdcmmembercommand	1217
gdcmmembercommand	1032
gdcmmembercommand	1033
gdcmmembercommand	265
gdcmmembercommand	1035

gdcm::network::RoleSelectionSub	1047
gdcm::SerieHelper	1108
gdcm::Series	1113
gdcm::network::ServiceClassApplicationInformation	1114
gdcm::SHA1	1124
gdcm::SimpleSubjectWatcher	1131
gdcm::MrProtocol::Slice	1134
gdcm::MrProtocol::SliceArray	1135
gdcm::SmartPointer< ObjectType >	1136
gdcm::network::SOPClassExtendedNegociationSub	1140
gdcm::SOPClassUIDToIOD	1142
gdcm::Sorter	1143
gdcm::IPPSorter	714
gdcm::Spacing	1148
gdcm::Spectroscopy	1150
gdcm::SplitMosaicFilter	1151
gdcm::static_assert_test< x >	1156
gdcm::STATIC_ASSERTION_FAILURE< x >	1157
gdcm::STATIC_ASSERTION_FAILURE< true >	1157
gdcm::StreamImageReader	1158
gdcm::StreamImageWriter	1162
String<'\\', 64 >	
gdcm::LO	773
gdcm::StringFilter	1193
gdcm::Study	1197
gdcm::SurfaceHelper	1214
gdcm::SwapCode	1225
gdcm::SwapperDoOp	1228
gdcm::SwapperNoOp	1229
gdcm::System	1229
gdcm::Table	1236
gdcm::TableEntry	1239
gdcm::TableReader	1240
gdcm::XMLDictReader	1564
gdcm::XMLPrivateDictReader	1571
gdcm::network::TableRow	1244
gdcm::Tag	1245
gdcm::PrivateTag	990
gdcm::TagPath	1255
gdcm::Testing	1257
gdcm::Trace	1264
gdcm::TransferSyntax	1269
gdcm::network::TransferSyntaxSub	1275
gdcm::network::Transition	1277
gdcm::Type	1279
gdcm::UI	1281
gdcm::UIDGenerator	1282
gdcm::UIDs	1285
gdcm::network::ULAction	1329
gdcm::network::ULActionAA1	1333
gdcm::network::ULActionAA2	1334
gdcm::network::ULActionAA3	1335
gdcm::network::ULActionAA4	1337

gdcm::network::ULActionAA5	1338
gdcm::network::ULActionAA6	1339
gdcm::network::ULActionAA7	1341
gdcm::network::ULActionAA8	1342
gdcm::network::ULActionAE1	1343
gdcm::network::ULActionAE2	1345
gdcm::network::ULActionAE3	1346
gdcm::network::ULActionAE4	1347
gdcm::network::ULActionAE5	1349
gdcm::network::ULActionAE6	1350
gdcm::network::ULActionAE7	1351
gdcm::network::ULActionAE8	1353
gdcm::network::ULActionAR1	1354
gdcm::network::ULActionAR10	1355
gdcm::network::ULActionAR2	1357
gdcm::network::ULActionAR3	1358
gdcm::network::ULActionAR4	1359
gdcm::network::ULActionAR5	1361
gdcm::network::ULActionAR6	1362
gdcm::network::ULActionAR7	1363
gdcm::network::ULActionAR8	1365
gdcm::network::ULActionAR9	1366
gdcm::network::ULActionDT1	1367
gdcm::network::ULActionDT2	1369
gdcm::network::ULConnection	1373
gdcm::network::ULConnectionCallback	1378
gdcm::network::ULBasicCallback	1370
gdcm::network::ULWritingCallback	1393
gdcm::network::ULConnectionInfo	1380
gdcm::network::ULEvent	1390
gdcm::network::ULTransitionTable	1391
gdcm::Unpacker12Bits	1403
gdcm::Usage	1404
gdcm::network::UserInformation	1408
gdcm::UUIDGenerator	1410
gdcm::Validate	1411
gdcm::ValueIO< TDE, TSwap, TType >	1417
gdcm::MrProtocol::Vector3	1418
gdcm::Version	1419
gdcm::VL	1420
gdcm::VM	1424
gdcm::VMToLength< T >	1430
gdcm::VR	1430
gdcm::VRToEncoding< T >	1441
gdcm::VRToType< T >	1441
gdcm::VRVLSIZE< T >	1442
gdcm::VRVLSIZE< 0 >	1442
gdcm::VRVLSIZE< 1 >	1444
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	1533
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	1529
vtkImageWriter	
vtkGDCMImageWriter	1474

vtkLookupTable	
vtkLookupTable16	1540
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties	1482
vtkMedicalImageReader2	
vtkGDCMImageReader	1445
vtkGDCMThreadedImageReader	1498
vtkGDCMImageReader2	1460
vtkObject	
vtkGDCMTesting	1494
vtkImageColorViewer	1510
vtkRTStructSetProperties	1543
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader	1485
vtkPolyDataWriter	
vtkGDCMPolyDataWriter	1490
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2	1503
vtkImageMapToColors16	1523
vtkImageRGBToYBR	1535
vtkImageYBRToRGB	1537
gdcm::Waveform	1553
gdcm::Writer	1558
gdcm::PixmapWriter	953
gdcm::ImageWriter	693
gdcm::SegmentWriter	1086
gdcm::SurfaceWriter	1221
gdcm::XMLPrinter	1567

Chapter 8

Class Index

8.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

gdcmm::network::AAabortPDU	
AAabortPDU	113
gdcmm::network::AAAssociateACPDU	
AAAssociateACPDU	116
gdcmm::network::AAAssociateRJPDU	
AAAssociateRJPDU	120
gdcmm::network::AAAssociateRQPDU	
AAAssociateRQPDU	122
gdcmm::AbortEvent	128
gdcmm::network::AbstractSyntax	
AbstractSyntax	129
gdcmm::AnonymizeEvent	
AnonymizeEvent	131
gdcmm::Anonymizer	
Anonymizer	135
gdcmm::AnyEvent	144
gdcmm::network::ApplicationContext	
ApplicationContext	145
gdcmm::ApplicationEntity	
ApplicationEntity	147
gdcmm::network::AReleaseRPPDU	
AReleaseRPPDU	149
gdcmm::network::AReleaseRQPDU	
AReleaseRQPDU	151
gdcmm::network::ARTIMTimer	
ARTIMTimer	154
gdcmm::ASN1	
Class for ASN1	155
gdcmm::network::AsynchronousOperationsWindowSub	
AsynchronousOperationsWindowSub	157

gdcmm::Attribute< Group, Element, TVR, TVM >	
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary	158
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >	168
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >	176
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >	181
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >	187
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >	194
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >	201
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >	207
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >	214
gdcmm::AudioCodec	
AudioCodec	219
gdcmm::Base64	
Class for Base64	222
gdcmm::network::BaseCompositeMessage	
BaseCompositeMessage	224
gdcmm::network::BaseNormalizedMessage	
BaseNormalizedMessage	226
gdcmm::network::BasePDU	
BasePDU	229
gdcmm::BaseQuery	
BaseQuery	231
gdcmm::BaseRootQuery	
BaseRootQuery	236
gdcmm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	241
gdcmm::BasicOffsetTable	
Class to represent a BasicOffsetTable	244
gdcmm::Bitmap	
Bitmap class	248
gdcmm::BitmapToBitmapFilter	
BitmapToBitmapFilter class	263
gdcmm::BoxRegion	
Class for manipulation box region	265
gdcmm::ByteBuffer	
ByteBuffer	270
gdcmm::ByteSwap< T >	
ByteSwap	271
gdcmm::ByteSwapFilter	
ByteSwapFilter	273
gdcmm::ByteValue	
Class to represent binary value (array of bytes)	275
gdcmm::CAPICryptoFactory	284
gdcmm::CAPICryptographicMessageSyntax	286
gdcmm::network::CEchoRQ	
CEchoRQ	289
gdcmm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	291
gdcmm::network::CFind	292
gdcmm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	293
gdcmm::network::CFindRQ	
CFindRQ	294

gdcM::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	296
gdcM::Cleaner	
Cleaner	297
gdcM::network::CMoveCancelRq	304
gdcM::network::CMoveRQ	
CMoveRQ	306
gdcM::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	307
gdcM::Codec	
Codec class	309
gdcM::Coder	
Coder	310
gdcM::CodeString	
CodeString	312
gdcM::Command	
Command superclass for callback/observer methods	316
gdcM::CommandDataSet	
Class to represent a Command DataSet	320
gdcM::network::CompositeMessageFactory	
CompositeMessageFactory	324
gdcM::CompositeNetworkFunctions	
Composite Network Functions	325
gdcM::ConstCharWrapper	
Do not use me	330
gdcM::CP246ExplicitDataElement	
Class to read/write a DataElement as CP246Explicit Data Element	331
gdcM::CryptoFactory	
Class to do handle the crypto factory	335
gdcM::CryptographicMessageSyntax	337
gdcM::CSAElement	
Class to represent a CSA Element	341
gdcM::CSAHeader	
Class for CSAHeader	348
gdcM::CSAHeaderDict	
Class to represent a map of CSAHeaderDictEntry	354
gdcM::CSAHeaderDictEntry	
Class to represent an Entry in the Dict	357
gdcM::CSAHeaderDictException	360
gdcM::network::CStoreRQ	
CStoreRQ	361
gdcM::network::CStoreRSP	
CStoreRSP this file defines the messages for the cecho action	363
gdcM::Curve	
Curve class to handle element 50xx,3000 Curve Data	364
gdcM::DataElement	
Class to represent a Data Element either Implicit or Explicit	369
gdcM::DataElementException	384
gdcM::DataEvent	
DataEvent	384
gdcM::DataSet	
Class to represent a Data Set (which contains Data Elements)	388
gdcM::DataSetEvent	
DataSetEvent	400

gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	404
gdcm::Decoder	
Decoder	405
gdcm::DefinedTerms	
Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor	406
gdcm::Defs	
FIXME I do not like the name 'Defs'	407
gdcm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	411
gdcm::DICOMDIR	
DICOMDIR class	415
gdcm::DICOMDIRGenerator	
DICOMDIRGenerator class	416
gdcm::Dict	
Class to represent a map of DictEntry	419
gdcm::DictConverter	
Class to convert a .dic file into something else:	424
gdcm::DictEntry	
Class to represent an Entry in the Dict	428
gdcm::DictPrinter	
DictPrinter class	432
gdcm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	435
gdcm::network::DIMSE	
DIMSE	439
gdcm::DirectionCosines	
Class to handle DirectionCosines	440
gdcm::Directory	
Class for manipulation directories	444
gdcm::DirectoryHelper	
DirectoryHelper	448
gdcm::DPath	
Class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA	450
gdcm::DummyValueGenerator	
Class for generating dummy value	452
gdcm::Dumper	
Codec class	453
gdcm::Element< TVR, TVM >	
Element class	456
gdcm::Element< TVR, VM::VM1_2 >	462
gdcm::Element< TVR, VM::VM1_n >	467
gdcm::Element< TVR, VM::VM2_2n >	474
gdcm::Element< TVR, VM::VM2_n >	480

gdcm::Element< TVR, VM::VM3_3n >	485
gdcm::Element< TVR, VM::VM3_4 >	492
gdcm::Element< TVR, VM::VM3_n >	497
gdcm::Element< VR::AS, VM::VM5 >	503
gdcm::Element< VR::OB, VM::VM1 >	507
gdcm::Element< VR::OW, VM::VM1 >	512
gdcm::ElementDisableCombinations< TVR, TVM >	
A class which is used to produce compile errors for an invalid combination of template parameters	517
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	518
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	519
gdcm::EmptyMaskGenerator	
EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series . This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory . .	520
gdcm::EncapsulatedDocument	
EncapsulatedDocument	522
gdcm::EncodingImplementation< T >	
EncodingImplementation	523
gdcm::EncodingImplementation< VR::VRASCII >	524
gdcm::EncodingImplementation< VR::VRBINARY >	526
gdcm::EndEvent	528
gdcm::EnumeratedValues	
Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:	529
gdcm::EquipmentManufacturer	530
gdcm::Event	
Superclass for callback/observer methods	532
gdcm::Exception	
Exception	535
gdcm::ExitEvent	538
gdcm::ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	539
gdcm::ExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element	543
gdcm::Fiducials	
Fiducials	547
gdcm::File	
DICOM File	548
gdcm::FileAnonymizer	
FileAnonymizer	553
gdcm::FileChangeTransferSyntax	
FileChangeTransferSyntax	557
gdcm::FileDecompressLookupTable	
FileDecompressLookupTable class	561
gdcm::FileDerivation	
FileDerivation class	564
gdcm::FileExplicitFilter	
FileExplicitFilter class	568
gdcm::FileMetaInformation	
Class to represent a File Meta Information	571
gdcm::Filename	
Class to manipulate file name's	581

gdcm::FileNameEvent	
FileNameEvent	584
gdcm::FilenameGenerator	
FilenameGenerator	588
gdcm::FileSet	591
gdcm::FileStreamer	
FileStreamer	593
gdcm::FileWithName	
FileWithName	599
gdcm::FindPatientRootQuery	
PatientRootQuery	602
gdcm::FindStudyRootQuery	
FindStudyRootQuery	606
gdcm::Fragment	
Class to represent a Fragment	610
gdcm::Global	
Global	615
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	618
gdcm::IconImageFilter	
IconImageFilter	621
gdcm::IconImageGenerator	
IconImageGenerator	624
gdcm::ignore_char	627
gdcm::Image	
Image	628
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class	637
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class	640
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class	645
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class	649
gdcm::ImageCodec	
ImageCodec	656
gdcm::ImageConverter	
Image Converter	668
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class	670
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	673
gdcm::ImageReader	
ImageReader	680
gdcm::ImageRegionReader	
ImageRegionReader	685
gdcm::ImageToImageFilter	
ImageToImageFilter class	690
gdcm::ImageWriter	
ImageWriter	693
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub	697
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub	699

gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub	699
gdcm::ImplicitDataElement	
Class to represent an Implicit VR Data Element	701
gdcm::InitializeEvent	705
gdcm::IOD	
Class for representing a IOD	706
gdcm::IODEntry	
Class for representing a IODEntry	709
gdcm::IODs	
Class for representing a IODs	711
gdcm::IPPSorter	
IPPSorter	714
gdcm::Item	
Class to represent an Item	720
gdcm::IterationEvent	726
gdcm::JPEG12Codec	
Class to do JPEG 12bits (lossy & lossless)	727
gdcm::JPEG16Codec	
Class to do JPEG 16bits (lossless)	733
gdcm::JPEG2000Codec	
Class to do JPEG 2000	738
gdcm::JPEG8Codec	
Class to do JPEG 8bits (lossy & lossless)	746
gdcm::JPEGCodec	
JPEG codec	751
gdcm::JPEGLSCodec	
JPEG-LS	760
gdcm::JSON	767
gdcm::KAKADUCodec	
KAKADUCodec	769
gdcm::LO	
LO	773
gdcm::LookupTable	
LookupTable class	777
gdcm::Scanner2::ltstr	784
gdcm::Scanner::ltstr	785
gdcm::StrictScanner2::ltstr	785
gdcm::StrictScanner::ltstr	786
gdcm::Macro	
Class for representing a Macro	787
gdcm::Macros	
Class for representing a Modules	789
gdcm::network::MaximumLengthSub	
MaximumLengthSub	791
gdcm::MD5	
Class for MD5	793
gdcm::MEC_MR3	
Class for MEC_MR3	794
gdcm::MediaStorage	
MediaStorage	795
gdcm::MemberCommand< T >	
Command subclass that calls a pointer to a member function	805

gdcmmesh::MeshPrimitive	
This class defines surface mesh primitives	812
gdcmmesh::ModalityPerformedProcedureStepCreateQuery	
ModalityPerformedProcedureStepCreateQuery	817
gdcmmesh::ModalityPerformedProcedureStepSetQuery	
ModalityPerformedProcedureStepSetQuery	821
gdcmmesh::ModifiedEvent	825
gdcmmesh::Module	
Class for representing a Module	826
gdcmmesh::ModuleEntry	
Class for representing a ModuleEntry	829
gdcmmesh::Modules	
Class for representing a Modules	833
gdcmmesh::MovePatientRootQuery	
MovePatientRootQuery	836
gdcmmesh::MoveStudyRootQuery	
MoveStudyRootQuery	840
gdcmmesh::MrProtocol	
Class for MrProtocol	844
gdcmmesh::network::NActionRQ	
NActionRQ	846
gdcmmesh::network::NActionRSP	
NActionRSP this file defines the messages for the NAction action	847
gdcmmesh::network::NCreateRQ	
NCreateRQ	849
gdcmmesh::network::NCreateRSP	
NCreateRSP this file defines the messages for the ncreate action	850
gdcmmesh::network::NDeleteRQ	
NDeleteRQ	852
gdcmmesh::network::NDeleteRSP	
NDeleteRSP this file defines the messages for the ndelete action	853
gdcmmesh::NestedModuleEntries	
Class for representing a NestedModuleEntries	855
gdcmmesh::network::NEventReportRQ	
NEventReportRQ	858
gdcmmesh::network::NEventReportRSP	
NEventReportRSP this file defines the messages for the neventreport action	859
gdcmmesh::network::NGetRQ	
NGetRQ	861
gdcmmesh::network::NGetRSP	
NGetRSP this file defines the messages for the nget action	862
gdcmmesh::NoEvent	864
gdcmmesh::network::NormalizedMessageFactory	865
gdcmmesh::NormalizedNetworkFunctions	
Normalized Network Functions	866
gdcmmesh::network::NSetRQ	
NSetRQ	869
gdcmmesh::network::NSetRSP	
NSetRSP this file defines the messages for the nset action	870
gdcmmesh::Object	
Object	872
gdcmmesh::OpenSSLCryptoFactory	875
gdcmmesh::OpenSSLCryptographicMessageSyntax	877
gdcmmesh::OpenSSLP7CryptoFactory	881

gdcm::OpenSSL7CryptographicMessageSyntax	883
gdcm::Orientation	
Class to handle Orientation	886
gdcm::Overlay	
Overlay class	889
gdcm::ParseException	
ParseException Standard exception handling object	898
gdcm::Parser	
Parser ala XML_Parser from expat (SAX)	900
gdcm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	904
gdcm::network::PDataTFPDU	
PDataTFPDU	905
gdcm::PDBelement	
Class to represent a PDB Element	908
gdcm::PDBHeader	
Class for PDBHeader	911
gdcm::PDFCodec	
PDFCodec class	914
gdcm::network::PDUFactory	
PDUFactory basically, given an initial byte, construct the	916
gdcm::PersonName	
PersonName class	920
gdcm::PGXCodec	
Class to do PGX	922
gdcm::PhotometricInterpretation	
Class to represent an PhotometricInterpretation	926
gdcm::PixelFormat	
PixelFormat	930
gdcm::Pixmap	
Pixmap class	939
gdcm::PixmapReader	
PixmapReader	946
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class	950
gdcm::PixmapWriter	
PixmapWriter	953
gdcm::PNMCodec	
Class to do PNM	958
gdcm::Preamble	
DICOM Preamble (Part 10)	963
gdcm::PresentationContext	
PresentationContext	967
gdcm::network::PresentationContextAC	
PresentationContextAC	970
gdcm::PresentationContextGenerator	
PresentationContextGenerator	972
gdcm::network::PresentationContextRQ	
PresentationContextRQ	975
gdcm::network::PresentationDataValue	
PresentationDataValue	979
gdcm::Printer	
Printer class	983

gdcmm::PrivateDict	
Private Dict	987
gdcmm::PrivateTag	
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element , Owner)	990
gdcmm::ProgressEvent	
ProgressEvent	995
gdcmm::PVRGCodec	
PVRGCodec	999
gdcmm::PythonFilter	
PythonFilter PythonFilter is the class that make gdcmm2.x looks more like gdcmm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language	1003
gdcmm::QueryBase	
QueryBase	1005
gdcmm::QueryFactory	
QueryFactory.h	1008
gdcmm::QueryImage	
QueryImage	1009
gdcmm::QueryPatient	
QueryPatient	1012
gdcmm::QuerySeries	
QuerySeries	1014
gdcmm::QueryStudy	
QueryStudy.h	1017
gdcmm::RAWCodec	
RAWCodec class	1019
gdcmm::Reader	
Reader ala DOM (Document Object Model)	1024
gdcmm::RealWorldValueMappingContent	1032
gdcmm::Region	
Class for manipulation region	1033
gdcmm::Rescaler	
Rescale class	1035
gdcmm::RLECodec	
Class to do RLE	1040
gdcmm::network::RoleSelectionSub	
RoleSelectionSub	1047
gdcmm::Scanner	
Scanner	1049
gdcmm::Scanner2	
Scanner2	1058
gdcmm::Segment	
This class defines a segment	1069
gdcmm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable class	1078
gdcmm::SegmentReader	
This class defines a segment reader	1082
gdcmm::SegmentWriter	
This class defines a segment writer	1086
gdcmm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	1092
gdcmm::SequenceOfItems	
Class to represent a Sequence Of Items	1099

gdcmm::SerieHelper	
SerieHelper	DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned 1108
gdcmm::Series	
Series 1113
gdcmm::network::ServiceClassApplicationInformation 1114
gdcmm::ServiceClassUser	
ServiceClassUser 1116
gdcmm::SHA1	
Class for SHA1 1124
gdcmm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function 1126
gdcmm::SimpleSubjectWatcher	
SimpleSubjectWatcher 1131
gdcmm::MrProtocol::Slice 1134
gdcmm::MrProtocol::SliceArray 1135
gdcmm::SmartPointer< ObjectType >	
Class for Smart Pointer 1136
gdcmm::network::SOPClassExtendedNegociationSub	
SOPClassExtendedNegociationSub 1140
gdcmm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD 1142
gdcmm::Sorter	
Sorter 1143
gdcmm::Spacing	
Class for Spacing 1148
gdcmm::Spectroscopy	
Spectroscopy class 1150
gdcmm::SplitMosaicFilter	
SplitMosaicFilter class 1151
gdcmm::StartEvent 1155
gdcmm::static_assert_test< x > 1156
gdcmm::STATIC_ASSERTION_FAILURE< x > 1157
gdcmm::STATIC_ASSERTION_FAILURE< true > 1157
gdcmm::StreamImageReader	
StreamImageReader 1158
gdcmm::StreamImageWriter	
StreamImageReader 1162
gdcmm::StrictScanner	
StrictScanner 1169
gdcmm::StrictScanner2	
StrictScanner2 1178
gdcmm::String< TDelimiter, TMaxLength, TPadChar >	
String 1189
gdcmm::StringFilter	
StringFilter 1193
gdcmm::Study	
Study 1197
gdcmm::Subject	
Subject 1198
gdcmm::Surface	
This class defines a SURFACE IE 1201
gdcmm::SurfaceHelper	
SurfaceHelper 1214

gdcm::SurfaceReader	
This class defines a SURFACE IE reader	1217
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer	1221
gdcm::SwapCode	
SwapCode representation	1225
gdcm::SwapperDoOp	1228
gdcm::SwapperNoOp	1229
gdcm::System	
Class to do system operation	1229
gdcm::Table	
Table	1236
gdcm::TableEntry	
TableEntry	1239
gdcm::TableReader	
Class for representing a TableReader	1240
gdcm::network::TableRow	1244
gdcm::Tag	
Class to represent a DICOM Data Element (Attribute) Tag (Group, Element)	1245
gdcm::TagPath	
Class to handle a path of tag	1255
gdcm::Testing	
Class for testing	1257
gdcm::Trace	
Trace	1264
gdcm::TransferSyntax	
Class to manipulate Transfer Syntax	1269
gdcm::network::TransferSyntaxSub	
TransferSyntaxSub	1275
gdcm::network::Transition	1277
gdcm::Type	
Type	1279
gdcm::UI	1281
gdcm::UIDGenerator	
Class for generating unique UID	1282
gdcm::UIDs	
All known uids	1285
gdcm::network::ULAction	
ULAction	1329
gdcm::network::ULActionAA1	1333
gdcm::network::ULActionAA2	1334
gdcm::network::ULActionAA3	1335
gdcm::network::ULActionAA4	1337
gdcm::network::ULActionAA5	1338
gdcm::network::ULActionAA6	1339
gdcm::network::ULActionAA7	1341
gdcm::network::ULActionAA8	1342
gdcm::network::ULActionAE1	1343
gdcm::network::ULActionAE2	1345
gdcm::network::ULActionAE3	1346
gdcm::network::ULActionAE4	1347
gdcm::network::ULActionAE5	1349
gdcm::network::ULActionAE6	1350
gdcm::network::ULActionAE7	1351

gdcmm::network::ULActionAE8	1353
gdcmm::network::ULActionAR1	1354
gdcmm::network::ULActionAR10	1355
gdcmm::network::ULActionAR2	1357
gdcmm::network::ULActionAR3	1358
gdcmm::network::ULActionAR4	1359
gdcmm::network::ULActionAR5	1361
gdcmm::network::ULActionAR6	1362
gdcmm::network::ULActionAR7	1363
gdcmm::network::ULActionAR8	1365
gdcmm::network::ULActionAR9	1366
gdcmm::network::ULActionDT1	1367
gdcmm::network::ULActionDT2	1369
gdcmm::network::ULBasicCallback	
ULBasicCallback	1370
gdcmm::network::ULConnection	
ULConnection	1373
gdcmm::network::ULConnectionCallback	1378
gdcmm::network::ULConnectionInfo	
ULConnectionInfo	1380
gdcmm::network::ULConnectionManager	
ULConnectionManager	1382
gdcmm::network::ULEvent	
ULEvent	1390
gdcmm::network::ULTransitionTable	
ULTransitionTable	The transition table of all the ULEvents, new ULActions, and ULStates1391
gdcmm::network::ULWritingCallback	1393
gdcmm::UNExplicitDataElement	
Class to read/write a DataElement as UNExplicit Data Element	1395
gdcmm::UNExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element	1399
gdcmm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	1403
gdcmm::Usage	
Usage	1404
gdcmm::UserEvent	1407
gdcmm::network::UserInformation	
UserInformation	1408
gdcmm::UUIDGenerator	
Class for generating unique UUID	1410
gdcmm::Validate	
Validate class	1411
gdcmm::Value	
Class to represent the value of a Data Element	1414
gdcmm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	1417
gdcmm::MrProtocol::Vector3	1418
gdcmm::Version	
Major/minor and build version	1419
gdcmm::VL	
Value Length	1420
gdcmm::VM	
Value Multiplicity	Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6
8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n	1424

gdcm::VMToLength< T >	1430
gdcm::VR	
VR class	1430
gdcm::VR16ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	1437
gdcm::VRToEncoding< T >	1441
gdcm::VRToType< T >	1441
gdcm::VRVLSize< T >	1442
gdcm::VRVLSize< 0 >	1442
gdcm::VRVLSize< 1 >	1444
vtkGDCMImageReader	1445
vtkGDCMImageReader2	1460
vtkGDCMImageWriter	1474
vtkGDCMMedicalImageProperties	1482
vtkGDCMPolyDataReader	1485
vtkGDCMPolyDataWriter	1490
vtkGDCMTesting	1494
vtkGDCMThreadedImageReader	1498
vtkGDCMThreadedImageReader2	1503
vtkImageColorViewer	1510
vtkImageMapToColors16	1523
vtkImageMapToWindowLevelColors2	1529
vtkImagePlanarComponentsToComponents	1533
vtkImageRGBToYBR	1535
vtkImageYBRToRGB	1537
vtkLookupTable16	1540
vtkRTStructSetProperties	1543
gdcm::Waveform	
Waveform class	1553
gdcm::WLMFindQuery	
PatientRootQuery	1554
gdcm::Writer	
Writer ala DOM (Document Object Model)	1558
gdcm::XMLDictReader	
Class for representing a XMLDictReader	1564
gdcm::XMLPrinter	1567
gdcm::XMLPrivateDictReader	
Class for representing a XMLPrivateDictReader	1571

Chapter 9

File Index

9.1 File List

Here is a list of all files with brief descriptions:

gdcmASN1.h	1575
gdcmBase64.h	1577
gdcmBoxRegion.h	1578
gdcmByteSwap.h	1579
gdcmCAPICryptoFactory.h	1581
gdcmCAPICryptographicMessageSyntax.h	1582
gdcmCommand.h	1585
gdcmCryptoFactory.h	1588
gdcmCryptographicMessageSyntax.h	1590
gdcmDataEvent.h	1592
gdcmDeflateStream.h	1594
gdcmDirectory.h	1594
gdcmDummyValueGenerator.h	1597
gdcmEvent.h	1598
gdcmException.h	1601
gdcmFilename.h	1605
gdcmFileNameEvent.h	1606
gdcmFilenameGenerator.h	1608
gdcmLegacyMacro.h	1609
gdcmMD5.h	1612
gdcmObject.h	1613
gdcmOpenSSLCryptoFactory.h	1616
gdcmOpenSSLCryptographicMessageSyntax.h	1617
gdcmOpenSSL7CryptoFactory.h	1619
gdcmOpenSSL7CryptographicMessageSyntax.h	1621
gdcmProgressEvent.h	1623
gdcmRegion.h	1624
gdcmSHA1.h	1627
gdcmSmartPointer.h	1628
gdcmStaticAssert.h	1630
gdcmString.h	1632

gdcmSubject.h	1636
gdcmSwapCode.h	1637
gdcmSwapper.h	1639
gdcmSystem.h	1642
gdcmTerminal.h	1644
gdcmTestDriver.h	1646
gdcmTesting.h	1647
gdcmTrace.h	1649
gdcmTypes.h	1655
gdcmUnpacker12Bits.h	1656
gdcmVersion.h	1657
gdcmWin32.h	1659
gdcmCSAHeaderDict.h	1660
gdcmCSAHeaderDictEntry.h	1663
gdcmDict.h	1666
gdcmDictConverter.h	1671
gdcmDictEntry.h	1673
gdcmDicts.h	1676
gdcmGlobal.h	1679
gdcmGroupDict.h	1681
gdcmSOPClassUIDToIOD.h	1683
gdcmUIDs.h	1684
gdcmAttribute.h	1697
gdcmBasicOffsetTable.h	1712
gdcmByteBuffer.h	1714
gdcmByteSwapFilter.h	1717
gdcmByteValue.h	1718
gdcmCodeString.h	1723
gdcmCP246ExplicitDataElement.h	1725
gdcmCSAElement.h	1726
gdcmCSAHeader.h	1729
gdcmDataElement.h	1732
gdcmDataSet.h	1735
gdcmDataSetEvent.h	1740
gdcmElement.h	1742
gdcmExplicitDataElement.h	1754
gdcmExplicitImplicitDataElement.h	1756
gdcmFile.h	1758
gdcmFileMetaInformation.h	1759
gdcmFileSet.h	1762
gdcmFragment.h	1764
gdcmImplicitDataElement.h	1769
gdcmItem.h	1770
gdcmLO.h	1776
gdcmMediaStorage.h	1777
gdcmMrProtocol.h	1781
gdcmParseException.h	1783
gdcmParser.h	1785
gdcmPDBElement.h	1788
gdcmPDBHeader.h	1790
gdcmPreamble.h	1792
gdcmPrivateTag.h	1794
gdcmReader.h	1796
gdcmSequenceOfFragments.h	1798

gdcmSequenceOfItems.h	1803
gdcmTag.h	1807
gdcmTagToVR.h	1812
gdcmTransferSyntax.h	1813
gdcmUNExplicitDataElement.h	1815
gdcmUNExplicitImplicitDataElement.h	1817
gdcmValue.h	1818
gdcmValueIO.h	1820
gdcmVL.h	1821
gdcmVM.h	1824
gdcmVR.h	1827
gdcmVR16ExplicitDataElement.h	1834
gdcmWriter.h	1836
gdcmDefinedTerms.h	1838
gdcmDefs.h	1839
gdcmEnumeratedValues.h	1842
gdcmIOD.h	1843
gdcmIODEntry.h	1846
gdcmIODs.h	1848
gdcmMacro.h	1851
gdcmMacroEntry.h	1854
gdcmMacros.h	1857
gdcmModule.h	1859
gdcmModuleEntry.h	1862
gdcmModules.h	1865
gdcmNestedModuleEntries.h	1867
gdcmPatient.h	1869
gdcmSeries.h	1871
gdcmStudy.h	1873
gdcmTable.h	1874
gdcmTableEntry.h	1876
gdcmTableReader.h	1878
gdcmType.h	1881
gdcmUsage.h	1883
gdcmXMLDictReader.h	1886
gdcmXMLPrivateDictReader.h	1888
gdcmAnonymizeEvent.h	1889
gdcmAnonymizer.h	1891
gdcmApplicationEntity.h	1894
gdcmAudioCodec.h	1895
gdcmBitmap.h	1896
gdcmBitmapToBitmapFilter.h	1900
gdcmCleaner.h	1901
gdcmCodec.h	1903
gdcmCoder.h	1904
gdcmConstCharWrapper.h	1906
gdcmCurve.h	1907
gdcmDataSetHelper.h	1910
gdcmDecoder.h	1911
gdcmDeltaEncodingCodec.h	1913
gdcmDICOMDIR.h	1914
gdcmDICOMDIRGenerator.h	1915
gdcmDictPrinter.h	1917
gdcmDirectionCosines.h	1918

gdcmDirectoryHelper.h	1920
gdcmDPath.h	1921
gdcmDumper.h	1924
gdcmEmptyMaskGenerator.h	1925
gdcmEncapsulatedDocument.h	1927
gdcmEquipmentManufacturer.h	1928
gdcmFiducials.h	1929
gdcmFileAnonymizer.h	1930
gdcmFileChangeTransferSyntax.h	1932
gdcmFileDecompressLookupTable.h	1934
gdcmFileDerivation.h	1936
gdcmFileExplicitFilter.h	1938
gdcmFileStreamer.h	1939
gdcmIconImage.h	1941
gdcmIconImageFilter.h	1943
gdcmIconImageGenerator.h	1945
gdcmImage.h	1946
gdcmImageApplyLookupTable.h	1949
gdcmImageChangePhotometricInterpretation.h	1950
gdcmImageChangePlanarConfiguration.h	1953
gdcmImageChangeTransferSyntax.h	1954
gdcmImageCodec.h	1956
gdcmImageConverter.h	1959
gdcmImageFragmentSplitter.h	1961
gdcmImageHelper.h	1962
gdcmImageReader.h	1964
gdcmImageRegionReader.h	1966
gdcmImageToImageFilter.h	1968
gdcmImageWriter.h	1969
gdcmIPPSorter.h	1971
gdcmJPEG12Codec.h	1973
gdcmJPEG16Codec.h	1974
gdcmJPEG2000Codec.h	1976
gdcmJPEG8Codec.h	1978
gdcmJPEGCodec.h	1979
gdcmJPEGLSCodec.h	1982
gdcmJSON.h	1983
gdcmKAKADUCodec.h	1985
gdcmLookupTable.h	1986
gdcmMEC_MR3.h	1989
gdcmMeshPrimitive.h	1990
gdcmOrientation.h	1993
gdcmOverlay.h	1994
gdcmPDFCodec.h	1997
gdcmPersonName.h	1998
gdcmPGXCodec.h	2000
gdcmPhotometricInterpretation.h	2001
gdcmPixelFormat.h	2004
gdcmPixmap.h	2008
gdcmPixmapReader.h	2010
gdcmPixmapToPixmapFilter.h	2012
gdcmPixmapWriter.h	2013
gdcmPNMCodec.h	2016
gdcmPrinter.h	2017

gdcmPVRGCodec.h	2020
gdcmRAWCodec.h	2021
gdcmRescaler.h	2022
gdcmRLECodec.h	2025
gdcmScanner.h	2026
gdcmScanner2.h	2029
gdcmSegment.h	2032
gdcmSegmentedPaletteColorLookupTable.h	2036
gdcmSegmentHelper.h	2037
gdcmSegmentReader.h	2039
gdcmSegmentWriter.h	2041
gdcmSerieHelper.h	2043
gdcmSimpleSubjectWatcher.h	2046
gdcmSorter.h	2048
gdcmSpacing.h	2051
gdcmSpectroscopy.h	2053
gdcmSplitMosaicFilter.h	2054
gdcmStreamImageReader.h	2056
gdcmStreamImageWriter.h	2057
gdcmStrictScanner.h	2059
gdcmStrictScanner2.h	2062
gdcmStringFilter.h	2065
gdcmSurface.h	2066
gdcmSurfaceHelper.h	2071
gdcmSurfaceReader.h	2073
gdcmSurfaceWriter.h	2075
gdcmTagPath.h	2076
gdcmUIDGenerator.h	2078
gdcmUUIDGenerator.h	2080
gdcmValidate.h	2081
gdcmWaveform.h	2082
gdcmXMLPrinter.h	2083
gdcmAAAbortPDU.h	2086
gdcmAAAssociateACPDU.h	2087
gdcmAAAssociateRJPDU.h	2090
gdcmAAAssociateRQPDU.h	2091
gdcmAbstractSyntax.h	2094
gdcmApplicationContext.h	2096
gdcmAReleaseRPPDU.h	2097
gdcmAReleaseRQPDU.h	2099
gdcmARTIMTimer.h	2101
gdcmAsynchronousOperationsWindowSub.h	2102
gdcmBaseCompositeMessage.h	2103
gdcmBaseNormalizedMessage.h	2105
gdcmBasePDU.h	2107
gdcmBaseQuery.h	2108
gdcmBaseRootQuery.h	2111
gdcmCEchoMessages.h	2113
gdcmCFindMessages.h	2114
gdcmCMoveMessages.h	2116
gdcmCommandDataSet.h	2118
gdcmCompositeMessageFactory.h	2119
gdcmCompositeNetworkFunctions.h	2121
gdcmCStoreMessages.h	2123

gdcmdIMSE.h	2124
gdcmdFindPatientRootQuery.h	2127
gdcmdFindStudyRootQuery.h	2128
gdcmdImplementationClassUIDSub.h	2130
gdcmdImplementationUIDSub.h	2132
gdcmdImplementationVersionNameSub.h	2133
gdcmdMaximumLengthSub.h	2135
gdcmdModalityPerformedProcedureStepCreateQuery.h	2137
gdcmdModalityPerformedProcedureStepSetQuery.h	2138
gdcmdMovePatientRootQuery.h	2140
gdcmdMoveStudyRootQuery.h	2141
gdcmdNActionMessages.h	2142
gdcmdNCreateMessages.h	2144
gdcmdNDeleteMessages.h	2145
gdcmdNetworkEvents.h	2146
gdcmdNetworkStateID.h	2148
gdcmdNEventReportMessages.h	2150
gdcmdNGetMessages.h	2152
gdcmdNormalizedMessageFactory.h	2153
gdcmdNormalizedNetworkFunctions.h	2154
gdcmdNSetMessages.h	2156
gdcmdPDataTFPDU.h	2157
gdcmdPDUFactory.h	2159
gdcmdPresentationContext.h	2161
gdcmdPresentationContextAC.h	2163
gdcmdPresentationContextGenerator.h	2165
gdcmdPresentationContextRQ.h	2166
gdcmdPresentationDataValue.h	2169
gdcmdQueryBase.h	2171
gdcmdQueryFactory.h	2173
gdcmdQueryImage.h	2175
gdcmdQueryPatient.h	2177
gdcmdQuerySeries.h	2179
gdcmdQueryStudy.h	2181
gdcmdRoleSelectionSub.h	2182
gdcmdServiceClassApplicationInformation.h	2184
gdcmdServiceClassUser.h	2185
gdcmdSOPClassExtendedNegociationSub.h	2188
gdcmdTransferSyntaxSub.h	2189
gdcmdULAction.h	2191
gdcmdULActionAA.h	2193
gdcmdULActionAE.h	2195
gdcmdULActionAR.h	2197
gdcmdULActionDT.h	2200
gdcmdULBasicCallback.h	2201
gdcmdULConnection.h	2202
gdcmdULConnectionCallback.h	2205
gdcmdULConnectionInfo.h	2206
gdcmdULConnectionManager.h	2208
gdcmdULEvent.h	2211
gdcmdULTransitionTable.h	2213
gdcmdULWritingCallback.h	2216
gdcmdUserInformation.h	2217
gdcmdWLMFindQuery.h	2219

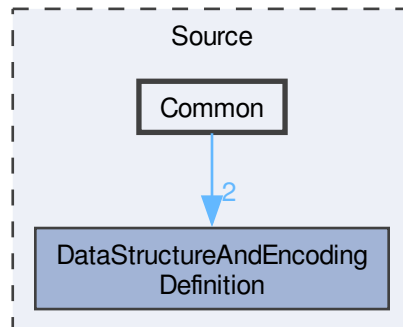
vtkGDCMImageReader.h	2220
vtkGDCMImageReader2.h	2226
vtkGDCMImageWriter.h	2231
vtkGDCMMedicalImageProperties.h	2234
vtkGDCMPolyDataReader.h	2240
vtkGDCMPolyDataWriter.h	2241
vtkGDCMTesting.h	2243
vtkGDCMThreadedImageReader.h	2245
vtkGDCMThreadedImageReader2.h	2247
vtkImageColorViewer.h	2249
vtkImageMapToColors16.h	2253
vtkImageMapToWindowLevelColors2.h	2255
vtkImagePlanarComponentsToComponents.h	2257
vtkImageRGBToYBR.h	2259
vtkImageYBRToRGB.h	2261
vtkLookupTable16.h	2262
vtkRTStructSetProperties.h	2264
gdcmPythonFilter.h	2266

Chapter 10

Directory Documentation

10.1 Common Directory Reference

Directory dependency graph for Common:



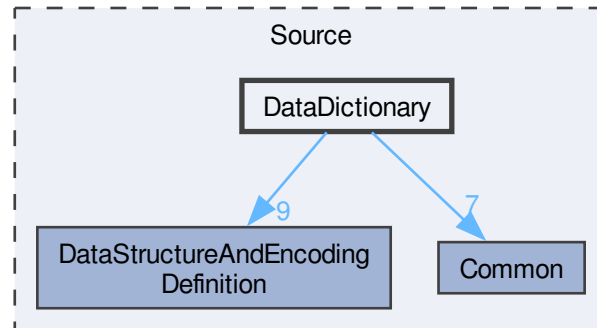
Files

- file [gdcmASN1.h](#)
- file [gdcmBase64.h](#)
- file [gdcmBoxRegion.h](#)
- file [gdcmByteSwap.h](#)
- file [gdcmCAPICryptoFactory.h](#)
- file [gdcmCAPICryptographicMessageSyntax.h](#)
- file [gdcmCommand.h](#)

- file [gdcryptoFactory.h](#)
- file [gdcryptoGraphicMessageSyntax.h](#)
- file [gdcDataEvent.h](#)
- file [gdcDeflateStream.h](#)
- file [gdcDirectory.h](#)
- file [gdcDummyValueGenerator.h](#)
- file [gdcEvent.h](#)
- file [gdcException.h](#)
- file [gdcFilename.h](#)
- file [gdcFileNameEvent.h](#)
- file [gdcFilenameGenerator.h](#)
- file [gdcLegacyMacro.h](#)
- file [gdcMD5.h](#)
- file [gdcObject.h](#)
- file [gdcOpenSSLCryptoFactory.h](#)
- file [gdcOpenSSLCryptoGraphicMessageSyntax.h](#)
- file [gdcOpenSSLP7CryptoFactory.h](#)
- file [gdcOpenSSLP7CryptoGraphicMessageSyntax.h](#)
- file [gdcProgressEvent.h](#)
- file [gdcRegion.h](#)
- file [gdcSHA1.h](#)
- file [gdcSmartPointer.h](#)
- file [gdcStaticAssert.h](#)
- file [gdcString.h](#)
- file [gdcSubject.h](#)
- file [gdcSwapCode.h](#)
- file [gdcSwapper.h](#)
- file [gdcSystem.h](#)
- file [gdcTerminal.h](#)
- file [gdcTestDriver.h](#)
- file [gdcTesting.h](#)
- file [gdcTrace.h](#)
- file [gdcTypes.h](#)
- file [gdcUnpacker12Bits.h](#)
- file [gdcVersion.h](#)
- file [gdcWin32.h](#)

10.2 DataDictionary Directory Reference

Directory dependency graph for DataDictionary:

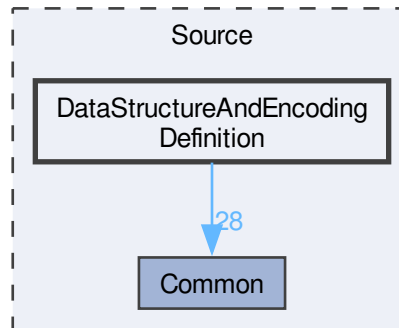


Files

- file [gdcmCSAHeaderDict.h](#)
- file [gdcmCSAHeaderDictEntry.h](#)
- file [gdcmDict.h](#)
- file [gdcmDictConverter.h](#)
- file [gdcmDictEntry.h](#)
- file [gdcmDicts.h](#)
- file [gdcmGlobal.h](#)
- file [gdcmGroupDict.h](#)
- file [gdcmSOPClassUIDToIOD.h](#)
- file [gdcmUIDs.h](#)

10.3 DataStructureAndEncodingDefinition Directory Reference

Directory dependency graph for DataStructureAndEncodingDefinition:



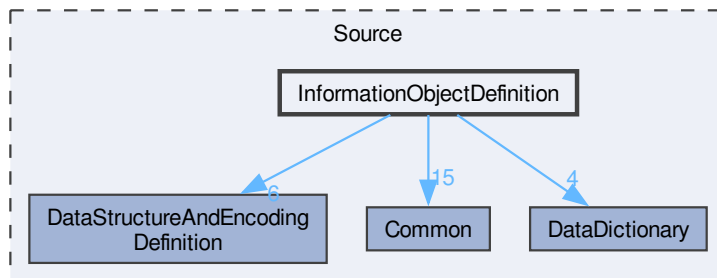
Files

- file [gdcAttribute.h](#)
- file [gdcBasicOffsetTable.h](#)
- file [gdcByteBuffer.h](#)
- file [gdcByteSwapFilter.h](#)
- file [gdcByteValue.h](#)
- file [gdcCodeString.h](#)
- file [gdcCP246ExplicitDataElement.h](#)
- file [gdcCSAElement.h](#)
- file [gdcCSAHeader.h](#)
- file [gdcDataElement.h](#)
- file [gdcDataSet.h](#)
- file [gdcDataSetEvent.h](#)
- file [gdcElement.h](#)
- file [gdcExplicitDataElement.h](#)
- file [gdcExplicitImplicitDataElement.h](#)
- file [gdcFile.h](#)
- file [gdcFileMetaInformation.h](#)
- file [gdcFileSet.h](#)
- file [gdcFragment.h](#)
- file [gdcImplicitDataElement.h](#)
- file [gdcItem.h](#)
- file [gdcLO.h](#)
- file [gdcMediaStorage.h](#)
- file [gdcMrProtocol.h](#)

- file [gdcmParseException.h](#)
- file [gdcmParser.h](#)
- file [gdcmPDBelement.h](#)
- file [gdcmPDBHeader.h](#)
- file [gdcmPreamble.h](#)
- file [gdcmPrivateTag.h](#)
- file [gdcmReader.h](#)
- file [gdcmSequenceOfFragments.h](#)
- file [gdcmSequenceOfItems.h](#)
- file [gdcmTag.h](#)
- file [gdcmTagToVR.h](#)
- file [gdcmTransferSyntax.h](#)
- file [gdcmUNExplicitDataElement.h](#)
- file [gdcmUNExplicitImplicitDataElement.h](#)
- file [gdcmValue.h](#)
- file [gdcmValueIO.h](#)
- file [gdcmVL.h](#)
- file [gdcmVM.h](#)
- file [gdcmVR.h](#)
- file [gdcmVR16ExplicitDataElement.h](#)
- file [gdcmWriter.h](#)

10.4 InformationObjectDefinition Directory Reference

Directory dependency graph for InformationObjectDefinition:



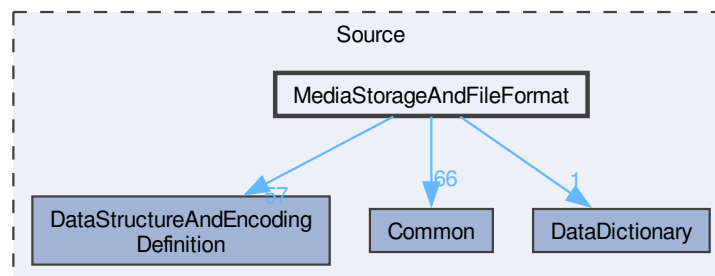
Files

- file [gdcmDefinedTerms.h](#)
- file [gdcmDefs.h](#)
- file [gdcmEnumeratedValues.h](#)
- file [gdcmIOD.h](#)

- file [gdcmIODEntry.h](#)
- file [gdcmIODs.h](#)
- file [gdcmMacro.h](#)
- file [gdcmMacroEntry.h](#)
- file [gdcmMacros.h](#)
- file [gdcmModule.h](#)
- file [gdcmModuleEntry.h](#)
- file [gdcmModules.h](#)
- file [gdcmNestedModuleEntries.h](#)
- file [gdcmPatient.h](#)
- file [gdcmSeries.h](#)
- file [gdcmStudy.h](#)
- file [gdcmTable.h](#)
- file [gdcmTableEntry.h](#)
- file [gdcmTableReader.h](#)
- file [gdcmType.h](#)
- file [gdcmUsage.h](#)
- file [gdcmXMLDictReader.h](#)
- file [gdcmXMLPrivateDictReader.h](#)

10.5 MediaStorageAndFileFormat Directory Reference

Directory dependency graph for MediaStorageAndFileFormat:



Files

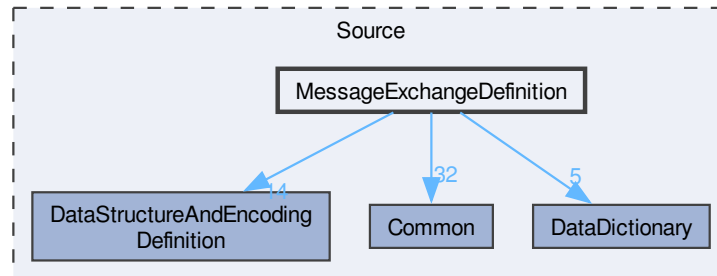
- file [gdcmAnonymizeEvent.h](#)
- file [gdcmAnonymizer.h](#)
- file [gdcmApplicationEntity.h](#)
- file [gdcmAudioCodec.h](#)
- file [gdcmBitmap.h](#)
- file [gdcmBitmapToBitmapFilter.h](#)

- file [gdcmCleaner.h](#)
- file [gdcmCodec.h](#)
- file [gdcmCoder.h](#)
- file [gdcmConstCharWrapper.h](#)
- file [gdcmCurve.h](#)
- file [gdcmDataSetHelper.h](#)
- file [gdcmDecoder.h](#)
- file [gdcmDeltaEncodingCodec.h](#)
- file [gdcmDICOMDIR.h](#)
- file [gdcmDICOMDIRGenerator.h](#)
- file [gdcmDictPrinter.h](#)
- file [gdcmDirectionCosines.h](#)
- file [gdcmDirectoryHelper.h](#)
- file [gdcmDPath.h](#)
- file [gdcmDumper.h](#)
- file [gdcmEmptyMaskGenerator.h](#)
- file [gdcmEncapsulatedDocument.h](#)
- file [gdcmEquipmentManufacturer.h](#)
- file [gdcmFiducials.h](#)
- file [gdcmFileAnonymizer.h](#)
- file [gdcmFileChangeTransferSyntax.h](#)
- file [gdcmFileDecompressLookupTable.h](#)
- file [gdcmFileDerivation.h](#)
- file [gdcmFileExplicitFilter.h](#)
- file [gdcmFileStreamer.h](#)
- file [gdcmIconImage.h](#)
- file [gdcmIconImageFilter.h](#)
- file [gdcmIconImageGenerator.h](#)
- file [gdcmImage.h](#)
- file [gdcmImageApplyLookupTable.h](#)
- file [gdcmImageChangePhotometricInterpretation.h](#)
- file [gdcmImageChangePlanarConfiguration.h](#)
- file [gdcmImageChangeTransferSyntax.h](#)
- file [gdcmImageCodec.h](#)
- file [gdcmImageConverter.h](#)
- file [gdcmImageFragmentSplitter.h](#)
- file [gdcmImageHelper.h](#)
- file [gdcmImageReader.h](#)
- file [gdcmImageRegionReader.h](#)
- file [gdcmImageToImageFilter.h](#)
- file [gdcmImageWriter.h](#)
- file [gdcmIPPSorter.h](#)
- file [gdcmJPEG12Codec.h](#)
- file [gdcmJPEG16Codec.h](#)
- file [gdcmJPEG2000Codec.h](#)
- file [gdcmJPEG8Codec.h](#)
- file [gdcmJPEGCodec.h](#)
- file [gdcmJPEGLSCCodec.h](#)
- file [gdcmJSON.h](#)
- file [gdcmKAKADUCodec.h](#)
- file [gdcmLookupTable.h](#)

- file [gdcmMEC_MR3.h](#)
- file [gdcmMeshPrimitive.h](#)
- file [gdcmOrientation.h](#)
- file [gdcmOverlay.h](#)
- file [gdcmPDFCodec.h](#)
- file [gdcmPersonName.h](#)
- file [gdcmPGXCodec.h](#)
- file [gdcmPhotometricInterpretation.h](#)
- file [gdcmPixelFormat.h](#)
- file [gdcmPixmap.h](#)
- file [gdcmPixmapReader.h](#)
- file [gdcmPixmapToPixmapFilter.h](#)
- file [gdcmPixmapWriter.h](#)
- file [gdcmPNMCodec.h](#)
- file [gdcmPrinter.h](#)
- file [gdcmPVRGCodec.h](#)
- file [gdcmRAWCodec.h](#)
- file [gdcmRescaler.h](#)
- file [gdcmRLECodec.h](#)
- file [gdcmScanner.h](#)
- file [gdcmScanner2.h](#)
- file [gdcmSegment.h](#)
- file [gdcmSegmentedPaletteColorLookupTable.h](#)
- file [gdcmSegmentHelper.h](#)
- file [gdcmSegmentReader.h](#)
- file [gdcmSegmentWriter.h](#)
- file [gdcmSerieHelper.h](#)
- file [gdcmSimpleSubjectWatcher.h](#)
- file [gdcmSorter.h](#)
- file [gdcmSpacing.h](#)
- file [gdcmSpectroscopy.h](#)
- file [gdcmSplitMosaicFilter.h](#)
- file [gdcmStreamImageReader.h](#)
- file [gdcmStreamImageWriter.h](#)
- file [gdcmStrictScanner.h](#)
- file [gdcmStrictScanner2.h](#)
- file [gdcmStringFilter.h](#)
- file [gdcmSurface.h](#)
- file [gdcmSurfaceHelper.h](#)
- file [gdcmSurfaceReader.h](#)
- file [gdcmSurfaceWriter.h](#)
- file [gdcmTagPath.h](#)
- file [gdcmUIDGenerator.h](#)
- file [gdcmUUIIDGenerator.h](#)
- file [gdcmValidate.h](#)
- file [gdcmWaveform.h](#)
- file [gdcmXMLPrinter.h](#)

10.6 MessageExchangeDefinition Directory Reference

Directory dependency graph for MessageExchangeDefinition:



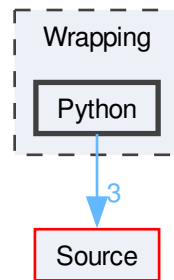
Files

- file [gdcmAAbortPDU.h](#)
- file [gdcmAAssociateACPDU.h](#)
- file [gdcmAAssociateRJPDU.h](#)
- file [gdcmAAssociateRQPDU.h](#)
- file [gdcmAbstractSyntax.h](#)
- file [gdcmApplicationContext.h](#)
- file [gdcmARReleaseRPPDU.h](#)
- file [gdcmARReleaseRQPDU.h](#)
- file [gdcmARTIMTimer.h](#)
- file [gdcmAsynchronousOperationsWindowSub.h](#)
- file [gdcmBaseCompositeMessage.h](#)
- file [gdcmBaseNormalizedMessage.h](#)
- file [gdcmBasePDU.h](#)
- file [gdcmBaseQuery.h](#)
- file [gdcmBaseRootQuery.h](#)
- file [gdcmCEchoMessages.h](#)
- file [gdcmCFindMessages.h](#)
- file [gdcmCMoveMessages.h](#)
- file [gdcmCommandDataSet.h](#)
- file [gdcmCompositeMessageFactory.h](#)
- file [gdcmCompositeNetworkFunctions.h](#)
- file [gdcmCStoreMessages.h](#)
- file [gdcmDIMSE.h](#)
- file [gdcmFindPatientRootQuery.h](#)
- file [gdcmFindStudyRootQuery.h](#)
- file [gdcmImplementationClassUIDSub.h](#)
- file [gdcmImplementationUIDSub.h](#)

- file [gdcmImplementationVersionNameSub.h](#)
- file [gdcmMaximumLengthSub.h](#)
- file [gdcmModalityPerformedProcedureStepCreateQuery.h](#)
- file [gdcmModalityPerformedProcedureStepSetQuery.h](#)
- file [gdcmMovePatientRootQuery.h](#)
- file [gdcmMoveStudyRootQuery.h](#)
- file [gdcmNActionMessages.h](#)
- file [gdcmNCreateMessages.h](#)
- file [gdcmNDeleteMessages.h](#)
- file [gdcmNetworkEvents.h](#)
- file [gdcmNetworkStateID.h](#)
- file [gdcmNEventReportMessages.h](#)
- file [gdcmNGetMessages.h](#)
- file [gdcmNormalizedMessageFactory.h](#)
- file [gdcmNormalizedNetworkFunctions.h](#)
- file [gdcmNSetMessages.h](#)
- file [gdcmPDataTFPDU.h](#)
- file [gdcmPDUFactory.h](#)
- file [gdcmPresentationContext.h](#)
- file [gdcmPresentationContextAC.h](#)
- file [gdcmPresentationContextGenerator.h](#)
- file [gdcmPresentationContextRQ.h](#)
- file [gdcmPresentationDataValue.h](#)
- file [gdcmQueryBase.h](#)
- file [gdcmQueryFactory.h](#)
- file [gdcmQueryImage.h](#)
- file [gdcmQueryPatient.h](#)
- file [gdcmQuerySeries.h](#)
- file [gdcmQueryStudy.h](#)
- file [gdcmRoleSelectionSub.h](#)
- file [gdcmServiceClassApplicationInformation.h](#)
- file [gdcmServiceClassUser.h](#)
- file [gdcmSOPClassExtendedNegociationSub.h](#)
- file [gdcmTransferSyntaxSub.h](#)
- file [gdcmULAction.h](#)
- file [gdcmULActionAA.h](#)
- file [gdcmULActionAE.h](#)
- file [gdcmULActionAR.h](#)
- file [gdcmULActionDT.h](#)
- file [gdcmULBasicCallback.h](#)
- file [gdcmULConnection.h](#)
- file [gdcmULConnectionCallback.h](#)
- file [gdcmULConnectionInfo.h](#)
- file [gdcmULConnectionManager.h](#)
- file [gdcmULEvent.h](#)
- file [gdcmULTransitionTable.h](#)
- file [gdcmULWritingCallback.h](#)
- file [gdcmUserInformation.h](#)
- file [gdcmWLMFindQuery.h](#)

10.7 Python Directory Reference

Directory dependency graph for Python:

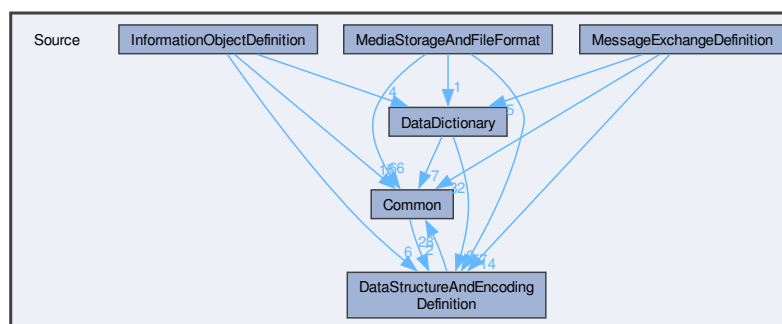


Files

- file [gdcmPythonFilter.h](#)

10.8 Source Directory Reference

Directory dependency graph for Source:



Directories

- directory [Common](#)
- directory [DataDictionary](#)
- directory [DataStructureAndEncodingDefinition](#)
- directory [InformationObjectDefinition](#)
- directory [MediaStorageAndFileFormat](#)
- directory [MessageExchangeDefinition](#)

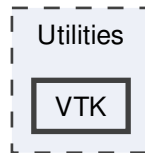
10.9 Utilities Directory Reference

Directories

- directory [VTK](#)

10.10 VTK Directory Reference

Directory dependency graph for VTK:

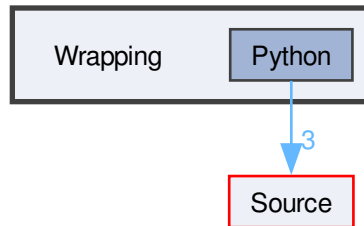


Files

- file [vtkGDCMImageReader.h](#)
- file [vtkGDCMImageReader2.h](#)
- file [vtkGDCMImageWriter.h](#)
- file [vtkGDCMMedicalImageProperties.h](#)
- file [vtkGDCMPolyDataReader.h](#)
- file [vtkGDCMPolyDataWriter.h](#)
- file [vtkGDCMTesting.h](#)
- file [vtkGDCMThreadedImageReader.h](#)
- file [vtkGDCMThreadedImageReader2.h](#)
- file [vtkImageColorViewer.h](#)
- file [vtkImageMapToColors16.h](#)
- file [vtkImageMapToWindowLevelColors2.h](#)
- file [vtkImagePlanarComponentsToComponents.h](#)
- file [vtkImageRGBToYBR.h](#)
- file [vtkImageYBRToRGB.h](#)
- file [vtkLookupTable16.h](#)
- file [vtkRTStructSetProperties.h](#)

10.11 Wrapping Directory Reference

Directory dependency graph for Wrapping:



Directories

- directory [Python](#)

Chapter 11

Namespace Documentation

11.1 gdcmm Namespace Reference

Namespaces

- namespace [network](#)
- namespace [SegmentHelper](#)
- namespace [terminal](#)

Class for Terminal.

Classes

- class [AbortEvent](#)
- class [AnonymizeEvent](#)
[AnonymizeEvent](#).
- class [Anonymizer](#)
[Anonymizer](#).
- class [AnyEvent](#)
- class [ApplicationEntity](#)
[ApplicationEntity](#).
- class [ASN1](#)
Class for [ASN1](#).
- class [Attribute](#)
[Attribute](#) class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.
- class [Attribute< Group, Element, TVR, VM::VM1_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM1_3_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM1_8_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM1_n_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM2_2n_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM2_n_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM3_3n_ >](#)
- class [Attribute< Group, Element, TVR, VM::VM3_n_ >](#)

- class [AudioCodec](#)
[AudioCodec](#).
- class [Base64](#)
Class for [Base64](#).
- class [BaseQuery](#)
[BaseQuery](#).
- class [BaseRootQuery](#)
[BaseRootQuery](#).
- class [BasicOffsetTable](#)
Class to represent a [BasicOffsetTable](#).
- class [Bitmap](#)
[Bitmap](#) class.
- class [BitmapToBitmapFilter](#)
[BitmapToBitmapFilter](#) class.
- class [BoxRegion](#)
Class for manipulation box region.
- class [ByteBuffer](#)
[ByteBuffer](#).
- class [ByteSwap](#)
[ByteSwap](#).
- class [ByteSwapFilter](#)
[ByteSwapFilter](#).
- class [ByteValue](#)
Class to represent binary value (array of bytes).
- class [CAPICryptoFactory](#)
- class [CAPICryptographicMessageSyntax](#)
- class [Cleaner](#)
[Cleaner](#).
- class [Codec](#)
[Codec](#) class.
- class [Coder](#)
[Coder](#).
- class [CodeString](#)
[CodeString](#).
- class [Command](#)
[Command](#) superclass for callback/observer methods.
- class [CommandDataSet](#)
Class to represent a [Command DataSet](#).
- class [CompositeNetworkFunctions](#)
Composite Network Functions.
- class [ConstCharWrapper](#)
Do not use me.
- class [CP246ExplicitDataElement](#)
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).
- class [CryptoFactory](#)
Class to do handle the crypto factory.
- class [CryptographicMessageSyntax](#)

- class [CSAElement](#)
Class to represent a CSA [Element](#).
- class [CSAHeader](#)
Class for [CSAHeader](#).
- class [CSAHeaderDict](#)
Class to represent a map of [CSAHeaderDictEntry](#).
- class [CSAHeaderDictEntry](#)
Class to represent an Entry in the [Dict](#).
- class [CSAHeaderDictException](#)
- class [Curve](#)
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.
- class [DataElement](#)
Class to represent a Data [Element](#) either Implicit or Explicit.
- class [DataElementException](#)
- class [DataEvent](#)
[DataEvent](#).
- class [DataSet](#)
Class to represent a Data Set (which contains Data Elements).
- class [DataSetEvent](#)
[DataSetEvent](#).
- class [DataSetHelper](#)
[DataSetHelper](#) (internal class, not intended for user level).
- class [Decoder](#)
[Decoder](#).
- class [DefinedTerms](#)
Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.
- class [Defs](#)
FIXME I do not like the name '[Defs](#)'.
- class [DeltaEncodingCodec](#)
[DeltaEncodingCodec](#) compression used by some private vendor.
- class [DICOMDIR](#)
[DICOMDIR](#) class.
- class [DICOMDIRGenerator](#)
[DICOMDIRGenerator](#) class.
- class [Dict](#)
Class to represent a map of [DictEntry](#).
- class [DictConverter](#)
Class to convert a .dic file into something else:
- class [DictEntry](#)
Class to represent an Entry in the [Dict](#).
- class [DictPrinter](#)

- [DictPrinter](#) class.
- class [Dicts](#)
 - Class to manipulate the sum of knowledge (all the dict user load).
- class [DirectionCosines](#)
 - class to handle [DirectionCosines](#)
- class [Directory](#)
 - Class for manipulation directories.
- class [DirectoryHelper](#)
 - [DirectoryHelper](#).
- class [DPath](#)
 - class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA>
- class [DummyValueGenerator](#)
 - Class for generating dummy value.
- class [Dumper](#)
 - [Codec](#) class.
- class [Element](#)
 - [Element](#) class.
 - class [Element](#)< TVR, VM::VM1_2 >
 - class [Element](#)< TVR, VM::VM1_n >
 - class [Element](#)< TVR, VM::VM2_2n >
 - class [Element](#)< TVR, VM::VM2_n >
 - class [Element](#)< TVR, VM::VM3_3n >
 - class [Element](#)< TVR, VM::VM3_4 >
 - class [Element](#)< TVR, VM::VM3_n >
 - class [Element](#)< VR::AS, VM::VM5 >
 - class [Element](#)< VR::OB, VM::VM1 >
 - class [Element](#)< VR::OW, VM::VM1 >
 - class [ElementDisableCombinations](#)
 - A class which is used to produce compile errors for an invalid combination of template parameters.
 - class [ElementDisableCombinations](#)< VR::OB, VM::VM1_n >
 - class [ElementDisableCombinations](#)< VR::OW, VM::VM1_n >
 - class [EmptyMaskGenerator](#)
 - [EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.
 - class [EncapsulatedDocument](#)
 - [EncapsulatedDocument](#).
 - class [EncodingImplementation](#)
 - [EncodingImplementation](#).
 - class [EncodingImplementation](#)< VR::VRASCII >
 - class [EncodingImplementation](#)< VR::VRBINARY >
 - class [EndEvent](#)
 - class [EnumeratedValues](#)
 - [Element](#). A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:
 - class [EquipmentManufacturer](#)

- class [Event](#)
superclass for callback/observer methods
- class [Exception](#)
[Exception](#).
- class [ExitEvent](#)
- class [ExplicitDataElement](#)
Class to read/write a [DataElement](#) as Explicit Data [Element](#).
- class [ExplicitImplicitDataElement](#)
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).
- class [Fiducials](#)
[Fiducials](#).
- class [File](#)
a DICOM [File](#)
- class [FileAnonymizer](#)
[FileAnonymizer](#).
- class [FileChangeTransferSyntax](#)
[FileChangeTransferSyntax](#).
- class [FileDecompressLookupTable](#)
[FileDecompressLookupTable](#) class.
- class [FileDerivation](#)
[FileDerivation](#) class.
- class [FileExplicitFilter](#)
[FileExplicitFilter](#) class.
- class [FileMetaInformation](#)
Class to represent a [File](#) Meta Information.
- class [Filename](#)
Class to manipulate file name's.
- class [FileNameEvent](#)
[FileNameEvent](#).
- class [FilenameGenerator](#)
[FilenameGenerator](#).
- class [FileSet](#)
- class [FileStreamer](#)
[FileStreamer](#).
- class [FileWithName](#)
[FileWithName](#).
- class [FindPatientRootQuery](#)
[PatientRootQuery](#).
- class [FindStudyRootQuery](#)
[FindStudyRootQuery](#).
- class [Fragment](#)
Class to represent a [Fragment](#).
- class [Global](#)
[Global](#).
- class [GroupDict](#)
Class to represent the mapping from group number to its abbreviation and name.
- class [IconImageFilter](#)

- [IconImageFilter](#).
- class [IconImageGenerator](#)
 - [IconImageGenerator](#).
- struct [ignore_char](#)
- class [Image](#)
 - [Image](#).
- class [ImageApplyLookupTable](#)
 - [ImageApplyLookupTable](#) class.
- class [ImageChangePhotometricInterpretation](#)
 - [ImageChangePhotometricInterpretation](#) class.
- class [ImageChangePlanarConfiguration](#)
 - [ImageChangePlanarConfiguration](#) class.
- class [ImageChangeTransferSyntax](#)
 - [ImageChangeTransferSyntax](#) class.
- class [ImageCodec](#)
 - [ImageCodec](#).
- class [ImageConverter](#)
 - [Image](#) Converter.
- class [ImageFragmentSplitter](#)
 - [ImageFragmentSplitter](#) class.
- class [ImageHelper](#)
 - [ImageHelper](#) (internal class, not intended for user level).
- class [ImageReader](#)
 - [ImageReader](#).
- class [ImageRegionReader](#)
 - [ImageRegionReader](#).
- class [ImageToImageFilter](#)
 - [ImageToImageFilter](#) class.
- class [ImageWriter](#)
 - [ImageWriter](#).
- class [ImplicitDataElement](#)
 - Class to represent an Implicit [VR](#) Data [Element](#).
- class [InitializeEvent](#)
- class [IOD](#)
 - Class for representing a [IOD](#).
- class [IODEntry](#)
 - Class for representing a [IODEntry](#).
- class [IODs](#)
 - Class for representing a [IODs](#).
- class [IPPSorter](#)
 - [IPPSorter](#).
- class [Item](#)
 - Class to represent an [Item](#).
- class [IterationEvent](#)
- class [JPEG12Codec](#)
 - Class to do JPEG 12bits (lossy & lossless).
- class [JPEG16Codec](#)

- Class to do JPEG 16bits (lossless).
- class [JPEG2000Codec](#)
 - Class to do JPEG 2000.
- class [JPEG8Codec](#)
 - Class to do JPEG 8bits (lossy & lossless).
- class [JPEGCodec](#)
 - JPEG codec.
- class [JPEGLSCodec](#)
 - JPEG-LS.
- class [JSON](#)
- class [KAKADUCodec](#)
 - KAKADUCodec.
- class [LO](#)
 - LO.
- class [LookupTable](#)
 - LookupTable class.
- class [Macro](#)
 - Class for representing a Macro.
- class [Macros](#)
 - Class for representing a Modules.
- class [MD5](#)
 - Class for MD5.
- class [MEC_MR3](#)
 - Class for MEC_MR3.
- class [MediaStorage](#)
 - MediaStorage.
- class [MemberCommand](#)
 - Command subclass that calls a pointer to a member function.
- class [MeshPrimitive](#)
 - This class defines surface mesh primitives.
- class [ModalityPerformedProcedureStepCreateQuery](#)
 - ModalityPerformedProcedureStepCreateQuery.
- class [ModalityPerformedProcedureStepSetQuery](#)
 - ModalityPerformedProcedureStepSetQuery.
- class [ModifiedEvent](#)
- class [Module](#)
 - Class for representing a Module.
- class [ModuleEntry](#)
 - Class for representing a ModuleEntry.
- class [Modules](#)
 - Class for representing a Modules.
- class [MovePatientRootQuery](#)
 - MovePatientRootQuery.
- class [MoveStudyRootQuery](#)
 - MoveStudyRootQuery.
- class [MrProtocol](#)
 - Class for MrProtocol.

- class [NestedModuleEntries](#)
Class for representing a [NestedModuleEntries](#).
- class [NoEvent](#)
- class [NormalizedNetworkFunctions](#)
Normalized Network Functions.
- class [Object](#)
[Object](#).
- class [OpenSSLCryptoFactory](#)
- class [OpenSSLCryptographicMessageSyntax](#)
- class [OpenSSLP7CryptoFactory](#)
- class [OpenSSLP7CryptographicMessageSyntax](#)
- class [Orientation](#)
class to handle [Orientation](#)
- class [Overlay](#)
[Overlay](#) class.
- class [ParseException](#)
[ParseException](#) Standard exception handling object.
- class [Parser](#)
[Parser](#) ala XML_Parser from expat (SAX).
- class [Patient](#)
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.
- class [PDBElement](#)
Class to represent a PDB [Element](#).
- class [PDBHeader](#)
Class for [PDBHeader](#).
- class [PDFCodec](#)
[PDFCodec](#) class.
- class [PersonName](#)
[PersonName](#) class.
- class [PGXCodec](#)
Class to do PGX.
- class [PhotometricInterpretation](#)
Class to represent an [PhotometricInterpretation](#).
- class [PixelFormat](#)
[PixelFormat](#).
- class [Pixmap](#)
[Pixmap](#) class.
- class [PixmapReader](#)
[PixmapReader](#).
- class [PixmapToPixmapFilter](#)
[PixmapToPixmapFilter](#) class.
- class [PixmapWriter](#)
[PixmapWriter](#).
- class [PNMCodec](#)
Class to do PNM.
- class [Preamble](#)
DICOM [Preamble](#) (Part 10).

- class [PresentationContext](#)
[PresentationContext](#).
- class [PresentationContextGenerator](#)
[PresentationContextGenerator](#).
- class [Printer](#)
[Printer](#) class.
- class [PrivateDict](#)
Private [Dict](#).
- class [PrivateTag](#)
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner).
- class [ProgressEvent](#)
[ProgressEvent](#).
- class [PVRGCodec](#)
[PVRGCodec](#).
- class [PythonFilter](#)
[PythonFilter](#) [PythonFilter](#) is the class that make gdcM2.x looks more like gdcM1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.
- class [QueryBase](#)
[QueryBase](#).
- class [QueryFactory](#)
[QueryFactory](#).h.
- class [QueryImage](#)
[QueryImage](#).
- class [QueryPatient](#)
[QueryPatient](#).
- class [QuerySeries](#)
[QuerySeries](#).
- class [QueryStudy](#)
[QueryStudy](#).h.
- class [RAWCodec](#)
[RAWCodec](#) class.
- class [Reader](#)
[Reader](#) ala DOM (Document [Object](#) Model).
- struct [RealWorldValueMappingContent](#)
- class [Region](#)
Class for manipulation region.
- class [Rescaler](#)
Rescale class.
- class [RLECodec](#)
Class to do RLE.
- class [Scanner](#)
[Scanner](#).
- class [Scanner2](#)
[Scanner2](#).
- class [Segment](#)
This class defines a segment.
- class [SegmentedPaletteColorLookupTable](#)

- [SegmentedPaletteColorLookupTable](#) class.
- class [SegmentReader](#)
 - This class defines a segment reader.
- class [SegmentWriter](#)
 - This class defines a segment writer.
- class [SequenceOfFragments](#)
 - Class to represent a Sequence Of Fragments.
- class [SequenceOfItems](#)
 - Class to represent a Sequence Of Items.
- class [SerieHelper](#)
 - [SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.
- class [Series](#)
 - [Series](#).
- class [ServiceClassUser](#)
 - [ServiceClassUser](#).
- class [SHA1](#)
 - Class for [SHA1](#).
- class [SimpleMemberCommand](#)
 - [Command](#) subclass that calls a pointer to a member function.
- class [SimpleSubjectWatcher](#)
 - [SimpleSubjectWatcher](#).
- class [SmartPointer](#)
 - Class for Smart Pointer.
- class [SOPClassUIDToIOD](#)
 - Class convert a class SOP Class UID into [IOD](#).
- class [Sorter](#)
 - [Sorter](#).
- class [Spacing](#)
 - Class for [Spacing](#).
- class [Spectroscopy](#)
 - [Spectroscopy](#) class.
- class [SplitMosaicFilter](#)
 - [SplitMosaicFilter](#) class.
- class [StartEvent](#)
- struct [static_assert_test](#)
- struct [STATIC_ASSERTION_FAILURE](#)
- struct [STATIC_ASSERTION_FAILURE< true >](#)
- class [StreamImageReader](#)
 - [StreamImageReader](#).
- class [StreamImageWriter](#)
 - [StreamImageReader](#).
- class [StrictScanner](#)
 - [StrictScanner](#).
- class [StrictScanner2](#)
 - [StrictScanner2](#).
- class [String](#)

- [String.](#)
- class [StringFilter](#)
 - [StringFilter.](#)
- class [Study](#)
 - [Study.](#)
- class [Subject](#)
 - [Subject.](#)
- class [Surface](#)
 - This class defines a SURFACE IE.
- class [SurfaceHelper](#)
 - [SurfaceHelper.](#)
- class [SurfaceReader](#)
 - This class defines a SURFACE IE reader.
- class [SurfaceWriter](#)
 - This class defines a SURFACE IE writer.
- class [SwapCode](#)
 - [SwapCode](#) representation.
- class [SwapperDoOp](#)
- class [SwapperNoOp](#)
- class [System](#)
 - Class to do system operation.
- class [Table](#)
 - [Table.](#)
- class [TableEntry](#)
 - [TableEntry.](#)
- class [TableReader](#)
 - Class for representing a [TableReader](#).
- class [Tag](#)
 - Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).
- class [TagPath](#)
 - class to handle a path of tag.
- class [Testing](#)
 - class for testing
- class [Trace](#)
 - [Trace.](#)
- class [TransferSyntax](#)
 - Class to manipulate Transfer Syntax.
- class [Type](#)
 - [Type.](#)
- struct [UI](#)
- class [UIDGenerator](#)
 - Class for generating unique UID.
- class [UIDs](#)
 - all known uids
- class [UNExplicitDataElement](#)
 - Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).
- class [UNExplicitImplicitDataElement](#)

- Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).
- class [Unpacker12Bits](#)
 - Pack/Unpack 12 bits pixel into 16bits.
- class [Usage](#)
 - [Usage](#).
- class [UserEvent](#)
- class [UUIDGenerator](#)
 - Class for generating unique UUID.
- class [Validate](#)
 - [Validate](#) class.
- class [Value](#)
 - Class to represent the value of a Data [Element](#).
- class [ValueIO](#)
 - Class to dispatch template calls.
- class [Version](#)
 - major/minor and build version
- class [VL](#)
 - [Value](#) Length.
- class [VM](#)
 - [Value](#) Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.
- struct [VMToLength](#)
- class [VR](#)
 - [VR](#) class.
- class [VR16ExplicitDataElement](#)
 - Class to read/write a [DataElement](#) as Explicit Data [Element](#).
- struct [VRToEncoding](#)
- struct [VRToType](#)
- class [VRVLSize](#)
- class [VRVLSize< 0 >](#)
- class [VRVLSize< 1 >](#)
- class [Waveform](#)
 - [Waveform](#) class.
- class [WLMFindQuery](#)
 - PatientRootQuery.
- class [Writer](#)
 - [Writer](#) ala DOM (Document [Object](#) Model).
- class [XMLDictReader](#)
 - Class for representing a [XMLDictReader](#).
- class [XMLPrinter](#)
- class [XMLPrivateDictReader](#)
 - Class for representing a [XMLPrivateDictReader](#).

Typedefs

- typedef [String](#)<'\\', 16 > [AECComp](#)
- typedef [String](#)<'\\', 64 > [ASComp](#)
- typedef bool(* [BOOL_FUNCTION_PFILE_PFILE_POINTER](#)) ([File](#) *, [File](#) *)
- typedef [String](#)<'\\', 16 > [CSCComp](#)
- typedef [String](#)<'\\', 64 > [DACComp](#)
- typedef [String](#)<'\\', 64 > [DTComp](#)
- typedef std::vector< [SmartPointer](#)< [FileWithName](#) > > [FileList](#)
- typedef [Bitmap](#) [IconImage](#)
- typedef [String](#)<'\\', 64 > [LOComp](#)
- typedef [String](#)<'\\', 64 > [LTComp](#)
- typedef [ModuleEntry](#) [MacroEntry](#)
- typedef [NestedModuleEntries](#) [NestedMacroEntries](#)
- typedef [String](#)<'\\', 64 > [PNComp](#)
- typedef [String](#)<'\\', 64 > [SHComp](#)
- typedef [String](#)<'\\', 64 > [STComp](#)
- typedef [String](#)<'\\', 16 > [TMComp](#)
- typedef [String](#)<'\\', 4294967294 > [UCComp](#)
- typedef [String](#)<'\\', 64, 0 > [UIComp](#)
- typedef [String](#)<'\\', 4294967294 > [URComp](#)
- typedef [String](#)<'\\', 64 > [UTComp](#)

Enumerations

- enum [CompOperators](#) {
[GDCM_EQUAL](#) = 0 ,
[GDCM_DIFFERENT](#) ,
[GDCM_GREATER](#) ,
[GDCM_GREATEROREQUAL](#) ,
[GDCM_LESS](#) ,
[GDCM_LESSCOREQUAL](#) }
- enum [ECharSet](#) {
[eLatin1](#) = 0 ,
[eLatin2](#) ,
[eLatin3](#) ,
[eLatin4](#) ,
[eCyrillic](#) ,
[eArabic](#) ,
[eGreek](#) ,
[eHebrew](#) ,
[eLatin5](#) ,
[eJapanese](#) ,
[eThai](#) ,
[eJapaneseKanjiMultibyte](#) ,
[eJapaneseSupplementaryKanjiMultibyte](#) ,
[eKoreanHangulHanjaMultibyte](#) ,
[eUTF8](#) ,
[eGB18030](#) }
- enum [ENQueryType](#) {
[eCreateMMPS](#) = 0 ,
[eSetMMPS](#) }

- enum [EQueryLevel](#) {
 [ePatient](#) = 0 ,
 [eStudy](#) = 1 ,
 [eSeries](#) = 2 ,
 [eImage](#) = 3 }
- enum [EQueryType](#) {
 [eFind](#) = 0 ,
 [eMove](#) ,
 [eWLMFind](#) }
- enum [ERootType](#) {
 [ePatientRootType](#) ,
 [eStudyRootType](#) }
- enum [LodModeType](#) {
 [LD_ALL](#) = 0x00000000 ,
 [LD_NOSEQ](#) = 0x00000001 ,
 [LD_NOSHADOW](#) = 0x00000002 ,
 [LD_NOSHADOWSEQ](#) = 0x00000004 }

Functions

- static int [add1](#) (char *buf, int n)
- [ignore_char](#) const [backslash](#) ('\\')
- template<typename T>
 static T [Clamp](#) (int v)
- static void [clean](#) (char *mant)
- static int [doround](#) (char *buf, unsigned int n)
- [VR::VRType](#) [GetVRFromTag](#) ([Tag](#) const &tag)
- bool [operator!=](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)
- bool [operator!=](#) (const [DataElement](#) &lhs, const [DataElement](#) &rhs)
- std::ostream & [operator<<](#) (std::ostream &_os, const [GroupDict](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [IOD](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [IODEntry](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [IODs](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Macro](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Macros](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [MediaStorage](#) &ms)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Module](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [ModuleEntry](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Modules](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [NestedModuleEntries](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Tag](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [TransferSyntax](#) &ts)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Type](#) &val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [UI](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [UIDs](#) &uid)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Usage](#) &val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [VM](#) &_val)
- std::ostream & [operator<<](#) (std::ostream &_os, const [VR](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [BasicOffsetTable](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- std::ostream & [operator<<](#) (std::ostream &os, const [CommandDataSet](#) &val)

- `std::ostream & operator<<` (`std::ostream &os`, `const CSAElement &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const CSAHeader &d`)
- `std::ostream & operator<<` (`std::ostream &os`, `const CSAHeaderDict &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const CSAHeaderDictEntry &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const DataElement &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const DataSet &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Dict &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const DictEntry &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Dicts &d`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Directory &d`)
- `std::ostream & operator<<` (`std::ostream &os`, `const DPath &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Event &e`)

Generic inserter operator for [Event](#) and its subclasses.

- `std::ostream & operator<<` (`std::ostream &os`, `const File &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const FileMetaInformation &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const FileSet &f`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Fragment &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Global &g`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Item &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const MrProtocol &d`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Object &obj`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Orientation &o`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PDSElement &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PDBHeader &d`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PhotometricInterpretation &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PixelFormat &pf`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Preamble &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PrivateDict &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const PrivateTag &val`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Region &r`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Scanner &s`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Scanner2 &s`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Sorter &s`)
- `std::ostream & operator<<` (`std::ostream &os`, `const StrictScanner &s`)
- `std::ostream & operator<<` (`std::ostream &os`, `const StrictScanner2 &s`)
- `std::ostream & operator<<` (`std::ostream &os`, `const SwapCode &sc`)
- `std::ostream & operator<<` (`std::ostream &os`, `const Version &v`)
- `std::ostream & operator<<` (`std::ostream &os`, `const VL &val`)
- `bool operator==` (`const CodeString &ref`, `const CodeString &cs`)
- `std::istream & operator>>` (`std::istream &_is`, `Tag &_val`)
- `std::istream & operator>>` (`std::istream &in`, `ignore_char const &ic`)
- `template<char TDelimiter, unsigned int TMaxLength, char TPadChar>`
`std::istream & operator>>` (`std::istream &is`, `String< TDelimiter, TMaxLength, TPadChar > &ms`)
- `template<typename T>`
`static int Round` (`T x`)
- `static int roundat` (`char *buf`, `size_t bufLen`, `unsigned int i`, `int iexp`)
- `template<typename Float>`
`static void x16printf` (`char *buf`, `int size`, `Float f`)

Variables

- static [Global GlobalInstance](#)

11.1.1 Detailed Description

This header defines the classes for the AA Actions, Association Abort Related Actions ([Table 9-9](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AE Actions, Association Establishment Related Actions ([Table 9-6](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AR Actions, Association Release Related Actions ([Table 9-8](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the DT Actions, Data Transfer Related Actions ([Table 9-8](#) of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

11.1.2 Typedef Documentation

11.1.2.1 AECComp

```
typedef String<'\\',16> gdcn::AECComp
```

11.1.2.2 ASComp

```
typedef String<'\\',64> gdcn::ASComp
```

11.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER

```
typedef bool(* gdcn::BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER) (File *, File *)
```

11.1.2.4 CSCComp

```
typedef String<'\\',16> gdcn::CSCComp
```

11.1.2.5 DComp

```
typedef String<'\\',64> gdcm::DComp
```

Examples

[TemplateEmptyImage.cxx](#).

11.1.2.6 DTComp

```
typedef String<'\\',64> gdcm::DTComp
```

11.1.2.7 FileList

```
typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList
```

11.1.2.8 IconImage

```
typedef Bitmap gdcm::IconImage
```

Examples

[ExtractIconFromFile.cxx](#).

11.1.2.9 LOComp

```
typedef String<'\\',64> gdcm::LOComp
```

11.1.2.10 LTComp

```
typedef String<'\\',64> gdcm::LTComp
```

11.1.2.11 MacroEntry

```
typedef ModuleEntry gdcm::MacroEntry
```

11.1.2.12 NestedMacroEntries

```
typedef NestedModuleEntries gdcm::NestedMacroEntries
```

11.1.2.13 PNComp

```
typedef String<'\\',64> gdcm::PNComp
```

11.1.2.14 SHComp

```
typedef String<'\\',64> gdcm::SHComp
```

11.1.2.15 STComp

```
typedef String<'\\',64> gdcm::STComp
```

11.1.2.16 TMComp

```
typedef String<'\\',16> gdcm::TMComp
```

Examples

[TemplateEmptyImage.cxx](#).

11.1.2.17 UCComp

```
typedef String<'\\',4294967294> gdcm::UCComp
```

11.1.2.18 UIComp

```
typedef String<'\\',64,0> gdcm::UIComp
```

11.1.2.19 URComp

```
typedef String<'\\',4294967294> gdcm::URComp
```

11.1.2.20 UTComp

```
typedef String<'\\',64> gdcm::UTComp
```

11.1.3 Enumeration Type Documentation

11.1.3.1 CompOperators

enum [gdcM::CompOperators](#)

Enumerator

GDCM_EQUAL	
GDCM_DIFFERENT	
GDCM_GREATER	
GDCM_GREATEROREQUAL	
GDCM_LESS	
GDCM_LESSCOREQUAL	

11.1.3.2 ECharSet

enum [gdcM::ECharSet](#)

The character sets enumerated in PS 3.3 2009 Annex C, section C.12.1.1.2 The resulting character set is stored in 0008,0005 The conversion to the data element is performed by the [QueryFactory](#) itself

Enumerator

eLatin1	
eLatin2	
eLatin3	
eLatin4	
eCyrillic	
eArabic	
eGreek	
eHebrew	
eLatin5	
eJapanese	
eThai	
eJapaneseKanjiMultibyte	
eJapaneseSupplementaryKanjiMultibyte	
eKoreanHangulHanjaMultibyte	
eUTF8	
eGB18030	

11.1.3.3 ENQueryType

enum [gdcm::ENQueryType](#)

Enumerator

eCreateMMPS	
eSetMMPS	

11.1.3.4 EQueryLevel

enum [gdcm::EQueryLevel](#)

Enumerator

ePatient	
eStudy	
eSeries	
eImage	

11.1.3.5 EQueryType

enum [gdcm::EQueryType](#)

Enumerator

eFind	
eMove	
eWLMFind	

11.1.3.6 ERootType

enum [gdcm::ERootType](#)

Enumerator

ePatientRootType	
eStudyRootType	

11.1.3.7 LodModeType

enum [gdcm::LodModeType](#)

Enumerator

LD_ALL	
LD_NOSEQ	
LD_NOSHADOW	
LD_NOSHADOWSEQ	

11.1.4 Function Documentation

11.1.4.1 add1()

```
int gdcm::add1 (
    char * buf,
    int n)    [static]
```

References [add1\(\)](#).

Referenced by [add1\(\)](#), and [doround\(\)](#).

11.1.4.2 backslash()

```
ignore\_char const gdcm::backslash (
    '\\')
```

References [backslash\(\)](#).

Referenced by [backslash\(\)](#), and [gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength\(\)](#).

11.1.4.3 Clamp()

```
template<typename T>
T gdcm::Clamp (
    int v)    [inline], [static]
```

References [gdcm_assert](#).

Referenced by [gdcm::ImageChangePhotometricInterpretation::RGB2YBR\(\)](#), and [gdcm::ImageChangePhotometricInterpretation::YBR2RGB\(\)](#).

11.1.4.4 `clean()`

```
void gdcmm::clean (
    char * mant)    [inline], [static]
```

References [clean\(\)](#).

Referenced by [clean\(\)](#), and [x16printf\(\)](#).

11.1.4.5 `doround()`

```
int gdcmm::doround (
    char * buf,
    unsigned int n)    [static]
```

References [add1\(\)](#), and [doround\(\)](#).

Referenced by [doround\(\)](#), and [roundat\(\)](#).

11.1.4.6 `GetVRFromTag()`

```
VR::VRType gdcmm::GetVRFromTag (
    Tag const & tag)
```

11.1.4.7 `operator”!=()` [1/2]

```
bool gdcmm::operator!= (
    const CodeString & ref,
    const CodeString & cs)    [inline]
```

Referenced by [operator!=\(\)](#).

11.1.4.8 `operator”!=()` [2/2]

```
bool gdcmm::operator!= (
    const DataElement & lhs,
    const DataElement & rhs)    [inline]
```

References [operator!=\(\)](#).

11.1.4.9 `operator<<()` [1/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & __os,
    const GroupDict & __val)    [inline]
```

11.1.4.10 operator<<() [2/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const IOD & _val) [inline]
```

11.1.4.11 operator<<() [3/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const IODEntry & _val) [inline]
```

11.1.4.12 operator<<() [4/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const IODs & _val) [inline]
```

11.1.4.13 operator<<() [5/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const Macro & _val) [inline]
```

11.1.4.14 operator<<() [6/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const Macros & _val) [inline]
```

11.1.4.15 operator<<() [7/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const MediaStorage & ms) [inline]
```

11.1.4.16 operator<<() [8/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const Module & _val) [inline]
```

11.1.4.17 operator<<() [9/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const ModuleEntry & _val) [inline]
```

11.1.4.18 operator<<() [10/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Modules & _val) [inline]
```

11.1.4.19 operator<<() [11/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const NestedModuleEntries & _val) [inline]
```

11.1.4.20 operator<<() [12/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Tag & _val) [inline]
```

11.1.4.21 operator<<() [13/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const TransferSyntax & ts) [inline]
```

11.1.4.22 operator<<() [14/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const Type & val) [inline]
```

11.1.4.23 operator<<() [15/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & _os,
    const UI & _val) [inline]
```

11.1.4.24 operator<<() [16/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const UIDs & uid) [inline]
```

References [gdcm::UIDs::GetName\(\)](#), and [gdcm::UIDs::GetString\(\)](#).

11.1.4.25 operator<<() [17/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const Usage & val) [inline]
```

11.1.4.26 operator<<() [18/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const VM & _val) [inline]
```

11.1.4.27 operator<<() [19/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & _os,  
    const VR & val) [inline]
```

11.1.4.28 operator<<() [20/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const BasicOffsetTable & val) [inline]
```

11.1.4.29 operator<<() [21/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const CodeString & str) [inline]
```

11.1.4.30 operator<<() [22/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const CommandDataSet & val) [inline]
```

11.1.4.31 operator<<() [23/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAElement & val) [inline]
```

11.1.4.32 operator<<() [24/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeader & d) [inline]
```

11.1.4.33 operator<<() [25/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeaderDict & val) [inline]
```

11.1.4.34 operator<<() [26/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const CSAHeaderDictEntry & val) [inline]
```

11.1.4.35 operator<<() [27/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const DataElement & val) [inline]
```

11.1.4.36 operator<<() [28/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const DataSet & val) [inline]
```

11.1.4.37 operator<<() [29/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Dict & val) [inline]
```

11.1.4.38 operator<<() [30/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const DictEntry & val) [inline]
```

11.1.4.39 operator<<() [31/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const Dicts & d) [inline]
```

11.1.4.40 operator<<() [32/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const Directory & d) [inline]
```

11.1.4.41 operator<<() [33/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const DPath & val) [inline]
```

11.1.4.42 operator<<() [34/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const Event & e) [inline]
```

Generic inserter operator for [Event](#) and its subclasses.

References [gdcm::Event::Print\(\)](#).

11.1.4.43 operator<<() [35/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const File & val) [inline]
```

11.1.4.44 operator<<() [36/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const FileMetaInformation & val) [inline]
```

11.1.4.45 operator<<() [37/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const FileSet & f) [inline]
```

11.1.4.46 operator<<() [38/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const Fragment & val) [inline]
```

11.1.4.47 operator<<() [39/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const Global & g) [inline]
```

11.1.4.48 operator<<() [40/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const Item & val) [inline]
```

11.1.4.49 operator<<() [41/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const MrProtocol & d) [inline]
```

11.1.4.50 operator<<() [42/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const Object & obj) [inline]
```

11.1.4.51 operator<<() [43/59]

```
std::ostream & gdcmm::operator<< (  
    std::ostream & os,  
    const Orientation & o) [inline]
```


11.1.4.52 operator<<() [44/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PDBElement & val) [inline]
```

11.1.4.53 operator<<() [45/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PDBHeader & d) [inline]
```

11.1.4.54 operator<<() [46/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PhotometricInterpretation & val) [inline]
```

11.1.4.55 operator<<() [47/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PixelFormat & pf) [inline]
```

11.1.4.56 operator<<() [48/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const Preamble & val) [inline]
```

11.1.4.57 operator<<() [49/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PrivateDict & val) [inline]
```

11.1.4.58 operator<<() [50/59]

```
std::ostream & gdcm::operator<< (  
    std::ostream & os,  
    const PrivateTag & val) [inline]
```

11.1.4.59 operator<<() [51/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Region & r) [inline]
```

References [gdcmm::Region::Print\(\)](#).

11.1.4.60 operator<<() [52/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Scanner & s) [inline]
```

11.1.4.61 operator<<() [53/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Scanner2 & s) [inline]
```

11.1.4.62 operator<<() [54/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const Sorter & s) [inline]
```

11.1.4.63 operator<<() [55/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const StrictScanner & s) [inline]
```

11.1.4.64 operator<<() [56/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const StrictScanner2 & s) [inline]
```

11.1.4.65 operator<<() [57/59]

```
std::ostream & gdcmm::operator<< (
    std::ostream & os,
    const SwapCode & sc) [inline]
```

11.1.4.66 operator<<() [58/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const Version & v) [inline]
```

11.1.4.67 operator<<() [59/59]

```
std::ostream & gdcm::operator<< (
    std::ostream & os,
    const VL & val) [inline]
```

11.1.4.68 operator==()

```
bool gdcm::operator==(
    const CodeString & ref,
    const CodeString & cs) [inline]
```

11.1.4.69 operator>>() [1/3]

```
std::istream & gdcm::operator>> (
    std::istream & __is,
    Tag & __val) [inline]
```

11.1.4.70 operator>>() [2/3]

```
std::istream & gdcm::operator>> (
    std::istream & in,
    ignore\_char const & ic) [inline]
```

References [gdcm::ignore_char::m_char](#).

11.1.4.71 operator>>() [3/3]

```
template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & gdcm::operator>> (
    std::istream & is,
    String< TDelimiter, TMaxLength, TPadChar > & ms) [inline]
```

11.1.4.72 Round()

```
template<typename T>
int gdcm::Round (
    T x) [inline], [static]
```

Referenced by [gdcm::ImageChangePhotometricInterpretation::RGB2YBR\(\)](#), and [gdcm::ImageChangePhotometricInterpretation::YBR2RGB\(\)](#).

11.1.4.73 roundat()

```
int gdcmm::roundat (
    char * buf,
    size_t buflen,
    unsigned int i,
    int iexp) [static]
```

References [doround\(\)](#), and [roundat\(\)](#).

Referenced by [roundat\(\)](#), and [x16printf\(\)](#).

11.1.4.74 x16printf()

```
template<typename Float>
void gdcmm::x16printf (
    char * buf,
    int size,
    Float f) [static]
```

References [clean\(\)](#), [roundat\(\)](#), and [x16printf\(\)](#).

Referenced by [gdcmm::EncodingImplementation< VR::VRASCII >::Write\(\)](#), and [x16printf\(\)](#).

11.1.5 Variable Documentation

11.1.5.1 GlobalInstance

[Global](#) gdcmm::GlobalInstance [static]

11.2 gdcmm::network Namespace Reference

Classes

- class [AAabortPDU](#)
[AAabortPDU](#).
- class [AAssociateACPDU](#)
[AAssociateACPDU](#).
- class [AAssociateRJPDU](#)
[AAssociateRJPDU](#).
- class [AAssociateRQPDU](#)
[AAssociateRQPDU](#).
- class [AbstractSyntax](#)
[AbstractSyntax](#).
- class [ApplicationContext](#)
[ApplicationContext](#).

- class [AReleaseRPPDU](#)
[AReleaseRPPDU](#).
- class [AReleaseRQPDU](#)
[AReleaseRQPDU](#).
- class [ARTIMTimer](#)
[ARTIMTimer](#).
- class [AsynchronousOperationsWindowSub](#)
[AsynchronousOperationsWindowSub](#).
- class [BaseCompositeMessage](#)
[BaseCompositeMessage](#).
- class [BaseNormalizedMessage](#)
[BaseNormalizedMessage](#).
- class [BasePDU](#)
[BasePDU](#).
- class [CEchoRQ](#)
[CEchoRQ](#).
- class [CEchoRSP](#)
[CEchoRSP](#) this file defines the messages for the cecho action.
- class [CFind](#)
- class [CFindCancelRQ](#)
[CFindCancelRQ](#) this file defines the messages for the cfind action.
- class [CFindRQ](#)
[CFindRQ](#).
- class [CFindRSP](#)
[CFindRSP](#) this file defines the messages for the cfind action.
- class [CMoveCancelRq](#)
- class [CMoveRQ](#)
[CMoveRQ](#).
- class [CMoveRSP](#)
[CMoveRSP](#) this file defines the messages for the cmove action.
- class [CompositeMessageFactory](#)
[CompositeMessageFactory](#).
- class [CStoreRQ](#)
[CStoreRQ](#).
- class [CStoreRSP](#)
[CStoreRSP](#) this file defines the messages for the cecho action.
- class [DIMSE](#)
[DIMSE](#).
- class [ImplementationClassUIDSub](#)
[ImplementationClassUIDSub](#).
- class [ImplementationUIDSub](#)
[ImplementationUIDSub](#).
- class [ImplementationVersionNameSub](#)
[ImplementationVersionNameSub](#).
- class [MaximumLengthSub](#)
[MaximumLengthSub](#).
- class [NActionRQ](#)

- [NActionRQ.](#)
- class [NActionRSP](#)
 - [NActionRSP](#) this file defines the messages for the NAction action.
- class [NCreateRQ](#)
 - [NCreateRQ.](#)
- class [NCreateRSP](#)
 - [NCreateRSP](#) this file defines the messages for the ncreate action.
- class [NDeleteRQ](#)
 - [NDeleteRQ.](#)
- class [NDeleteRSP](#)
 - [NDeleteRSP](#) this file defines the messages for the ndelete action.
- class [NEventReportRQ](#)
 - [NEventReportRQ.](#)
- class [NEventReportRSP](#)
 - [NEventReportRSP](#) this file defines the messages for the neventreport action.
- class [NGetRQ](#)
 - [NGetRQ.](#)
- class [NGetRSP](#)
 - [NGetRSP](#) this file defines the messages for the nget action.
- class [NormalizedMessageFactory](#)
- class [NSetRQ](#)
 - [NSetRQ.](#)
- class [NSetRSP](#)
 - [NSetRSP](#) this file defines the messages for the nset action.
- class [PDataTFPDU](#)
 - [PDataTFPDU.](#)
- class [PDUFactory](#)
 - [PDUFactory](#) basically, given an initial byte, construct the.
- class [PresentationContextAC](#)
 - [PresentationContextAC.](#)
- class [PresentationContextRQ](#)
 - [PresentationContextRQ.](#)
- class [PresentationDataValue](#)
 - [PresentationDataValue.](#)
- class [RoleSelectionSub](#)
 - [RoleSelectionSub.](#)
- class [ServiceClassApplicationInformation](#)
- class [SOPClassExtendedNegociationSub](#)
 - [SOPClassExtendedNegociationSub.](#)
- class [TableRow](#)
- class [TransferSyntaxSub](#)
 - [TransferSyntaxSub.](#)
- struct [Transition](#)
- class [ULAction](#)
 - [ULAction.](#)
- class [ULActionAA1](#)
- class [ULActionAA2](#)

- class [ULActionAA3](#)
- class [ULActionAA4](#)
- class [ULActionAA5](#)
- class [ULActionAA6](#)
- class [ULActionAA7](#)
- class [ULActionAA8](#)
- class [ULActionAE1](#)
- class [ULActionAE2](#)
- class [ULActionAE3](#)
- class [ULActionAE4](#)
- class [ULActionAE5](#)
- class [ULActionAE6](#)
- class [ULActionAE7](#)
- class [ULActionAE8](#)
- class [ULActionAR1](#)
- class [ULActionAR10](#)
- class [ULActionAR2](#)
- class [ULActionAR3](#)
- class [ULActionAR4](#)
- class [ULActionAR5](#)
- class [ULActionAR6](#)
- class [ULActionAR7](#)
- class [ULActionAR8](#)
- class [ULActionAR9](#)
- class [ULActionDT1](#)
- class [ULActionDT2](#)
- class [ULBasicCallback](#)
[ULBasicCallback.](#)
- class [ULConnection](#)
[ULConnection.](#)
- class [ULConnectionCallback](#)
- class [ULConnectionInfo](#)
[ULConnectionInfo.](#)
- class [ULConnectionManager](#)
[ULConnectionManager.](#)
- class [ULEvent](#)
[ULEvent.](#)
- class [ULTransitionTable](#)
[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.
- class [ULWritingCallback](#)
- class [UserInformation](#)
[UserInformation.](#)

Enumerations

- enum [EEventID](#) {
[eAASSOCIATERequestLocalUser](#) = 0 ,
[eTransportConnConfirmLocal](#) ,
[eASSOCIATE__ACPDUreceived](#) ,
[eASSOCIATE__RJPDUreceived](#) ,
[eTransportConnIndicLocal](#) ,
[eAASSOCIATE__RQPDUreceived](#) ,
[eAASSOCIATEResponseAccept](#) ,
[eAASSOCIATEResponseReject](#) ,
[ePDATArequest](#) ,
[ePDATATFPDU](#) ,
[eARELEASERequest](#) ,
[eARELEASE__RQPDUReceivedOpen](#) ,
[eARELEASE__RPPDUReceived](#) ,
[eARELEASEResponse](#) ,
[eAABORTRequest](#) ,
[eAABORTPDUReceivedOpen](#) ,
[eTransportConnectionClosed](#) ,
[eARTIMTimerExpired](#) ,
[eUnrecognizedPDUReceived](#) ,
[eEventDoesNotExist](#) }
- enum [EStateID](#) {
[eStaDoesNotExist](#) = 0 ,
[eSta1Idle](#) = 1 ,
[eSta2Open](#) = 2 ,
[eSta3WaitLocalAssoc](#) = 4 ,
[eSta4LocalAssocDone](#) = 8 ,
[eSta5WaitRemoteAssoc](#) = 16 ,
[eSta6TransferReady](#) = 32 ,
[eSta7WaitRelease](#) = 64 ,
[eSta8WaitLocalRelease](#) = 128 ,
[eSta9ReleaseCollisionRqLocal](#) = 256 ,
[eSta10ReleaseCollisionAc](#) = 512 ,
[eSta11ReleaseCollisionRq](#) = 1024 ,
[eSta12ReleaseCollisionAcLocal](#) = 2048 ,
[eSta13AwaitingClose](#) = 4096 }

Functions

- int [GetStateIndex](#) ([EStateID](#) inState)

Variables

- const int [cMaxEventID](#) = [eEventDoesNotExist](#)
- const int [cMaxStateID](#) = 13

11.2.1 Enumeration Type Documentation

11.2.1.1 EEventID

enum [gdcmm::network::EEventID](#)

Enumerator

eAASSOCIATERequestLocalUser	
eTransportConnConfirmLocal	
eASSOCIATE__ACPDUreceived	
eASSOCIATE__RJPDUreceived	
eTransportConnIndicLocal	
eAASSOCIATE__RQPDUreceived	
eAASSOCIATEResponseAccept	
eAASSOCIATEResponseReject	
ePDATArequest	
ePDATATFPDU	
eARELEASERequest	
eARELEASE__RQPDUReceivedOpen	
eARELEASE__RPPDUReceived	
eARELEASEResponse	
eAABORTRequest	
eAABORTPDUReceivedOpen	
eTransportConnectionClosed	
eARTIMTimerExpired	
eUnrecognizedPDUReceived	
eEventDoesNotExist	

11.2.1.2 EStateID

enum [gdcmm::network::EStateID](#)

Each network connection will be in a particular state at any given time. Those states have IDs as described in the standard ps3.8-2009, roughly 1-13. This enumeration lists those states. The actual ULState class will contain more information about transitions to other states.

name and date: 16 sept 2010 mmr

Enumerator

eStaDoesNotExist	
------------------	--

eSta1Idle	
eSta2Open	
eSta3WaitLocalAssoc	
eSta4LocalAssocDone	
eSta5WaitRemoteAssoc	
eSta6TransferReady	
eSta7WaitRelease	
eSta8WaitLocalRelease	
eSta9ReleaseCollisionRqLocal	
eSta10ReleaseCollisionAc	
eSta11ReleaseCollisionRq	
eSta12ReleaseCollisionAcLocal	
eSta13AwaitingClose	

11.2.2 Function Documentation

11.2.2.1 GetStateIndex()

```
int gdcn::network::GetStateIndex (
    EStateID inState) [inline]
```

References [eSta10ReleaseCollisionAc](#), [eSta11ReleaseCollisionRq](#), [eSta12ReleaseCollisionAcLocal](#), [eSta13AwaitingClose](#), [eSta1Idle](#), [eSta2Open](#), [eSta3WaitLocalAssoc](#), [eSta4LocalAssocDone](#), [eSta5WaitRemoteAssoc](#), [eSta6TransferReady](#), [eSta7WaitRelease](#), [eSta8WaitLocalRelease](#), [eSta9ReleaseCollisionRqLocal](#), and [eStaDoesNotExist](#).

11.2.3 Variable Documentation

11.2.3.1 cMaxEventID

```
const int gdcn::network::cMaxEventID = eEventDoesNotExist
```

11.2.3.2 cMaxStateID

```
const int gdcn::network::cMaxStateID = 13
```

Referenced by [gdcn::network::TableRow::TableRow\(\)](#), and [gdcn::network::TableRow::~~TableRow\(\)](#).

11.3 gdcn::SegmentHelper Namespace Reference

Classes

- struct [BasicCodedEntry](#)

This structure defines a basic coded entry with all of its attributes.

11.4 gdcmm::terminal Namespace Reference

Class for Terminal.

Enumerations

- enum `Attribute` {
 `reset` = 0 ,
 `bright` = 1 ,
 `dim` = 2 ,
 `underline` = 3 ,
 `blink` = 5 ,
 `reverse` = 7 ,
 `hidden` = 8 }
- enum `Color` {
 `black` = 0 ,
 `red` ,
 `green` ,
 `yellow` ,
 `blue` ,
 `magenta` ,
 `cyan` ,
 `white` }
- enum `Mode` {
 `CONSOLE` = 0 ,
 `VT100` }

Functions

- `GDCM_EXPORT` std::string `setattribute` (`Attribute` att)
- `GDCM_EXPORT` std::string `setbgcolor` (`Color` c)
- `GDCM_EXPORT` std::string `setfgcolor` (`Color` c)
- `GDCM_EXPORT` void `setmode` (`Mode` m)

11.4.1 Detailed Description

Class for Terminal.

Allow one to print in color in a shell

- support VT100 compatible shell
- win32 console

11.4.2 Enumeration Type Documentation

11.4.2.1 Attribute

enum [gdcmm::terminal::Attribute](#)

Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

11.4.2.2 Color

enum [gdcmm::terminal::Color](#)

Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

11.4.2.3 Mode

enum [gdcmm::terminal::Mode](#)

Enumerator

CONSOLE	
VT100	

11.4.3 Function Documentation

11.4.3.1 setattribute()

[GDCM_EXPORT](#) std::string gdcm::terminal::setattribute (
 [Attribute](#) att)

References [GDCM_EXPORT](#).

11.4.3.2 setbgcolor()

[GDCM_EXPORT](#) std::string gdcm::terminal::setbgcolor (
 [Color](#) c)

References [GDCM_EXPORT](#).

11.4.3.3 setfgcolor()

[GDCM_EXPORT](#) std::string gdcm::terminal::setfgcolor (
 [Color](#) c)

References [GDCM_EXPORT](#).

11.4.3.4 setmode()

[GDCM_EXPORT](#) void gdcm::terminal::setmode (
 [Mode](#) m)

References [GDCM_EXPORT](#).

Chapter 12

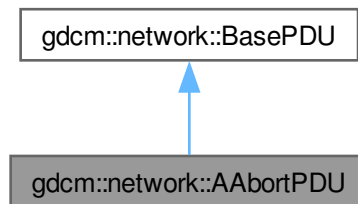
Class Documentation

12.1 gdcmm::network::AAabortPDU Class Reference

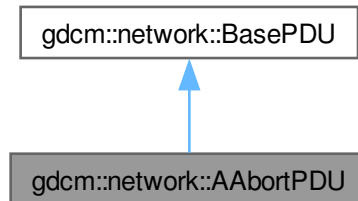
[AAabortPDU](#).

```
#include <gdcmmAAabortPDU.h>
```

Inheritance diagram for gdcmm::network::AAabortPDU:



Collaboration diagram for gdcmm::network::AAabortPDU:



Public Member Functions

- [AAbortPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- void [SetReason](#) (const uint8_t r)
- void [SetSource](#) (const uint8_t s)
- size_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

12.1.1 Detailed Description

[AAbortPDU](#).

[Table 9-26](#) A-ABORT PDU FIELDS

12.1.2 Constructor & Destructor Documentation

12.1.2.1 AAbortPDU()

`gdcm::network::AAbortPDU::AAbortPDU ()`

12.1.3 Member Function Documentation

12.1.3.1 IsLastFragment()

`bool gdcm::network::AAbortPDU::IsLastFragment () const` [\[inline\]](#), [\[override\]](#), [\[virtual\]](#)

Implements [gdcm::network::BasePDU](#).

12.1.3.2 Print()

`void gdcm::network::AAbortPDU::Print (`
`std::ostream & os) const` [\[override\]](#), [\[virtual\]](#)

Implements [gdcm::network::BasePDU](#).

12.1.3.3 Read()

```
std::istream & gdcm::network::AAAbortPDU::Read (  
    std::istream & is)    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.1.3.4 SetReason()

```
void gdcm::network::AAAbortPDU::SetReason (  
    const uint8_t r)
```

12.1.3.5 SetSource()

```
void gdcm::network::AAAbortPDU::SetSource (  
    const uint8_t s)
```

12.1.3.6 Size()

```
size_t gdcm::network::AAAbortPDU::Size () const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.1.3.7 Write()

```
const std::ostream & gdcm::network::AAAbortPDU::Write (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

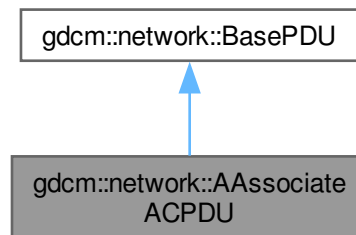
- [gdcmAAAbortPDU.h](#)

12.2 gdcm::network::AAssociateACPDU Class Reference

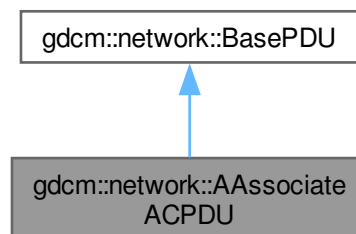
[AAssociateACPDU](#).

```
#include <gdcmAAssociateACPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateACPDU:



Collaboration diagram for gdcm::network::AAssociateACPDU:



Public Types

- typedef std::vector< [PresentationContextAC](#) >::size_type [SizeType](#)

Public Member Functions

- [AAssociateACPDU](#) ()
- void [AddPresentationContextAC](#) ([PresentationContextAC](#) const &pcac)
- [SizeType](#) [GetNumberOfPresentationContextAC](#) () const
- const [PresentationContextAC](#) & [GetPresentationContextAC](#) ([SizeType](#) i)
- const [UserInformation](#) & [GetUserInformation](#) () const
- void [InitFromRQ](#) ([AAssociateRQPDU](#) const &rqpdu)
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- [SizeType](#) [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

Protected Member Functions

- void [SetCalledAETitle](#) (const char calledaetitle[16])
- void [SetCallingAETitle](#) (const char callingaetitle[16])

Friends

- class [AAssociateRQPDU](#)

12.2.1 Detailed Description

[AAssociateACPDU](#).

[Table 9-17](#) ASSOCIATE-AC PDU fields

12.2.2 Member Typedef Documentation

12.2.2.1 SizeType

```
typedef std::vector<PresentationContextAC>::size_type gdcm::network::AAssociateACPDU::SizeType
```

12.2.3 Constructor & Destructor Documentation

12.2.3.1 AAssociateACPDU()

```
gdcm::network::AAssociateACPDU::AAssociateACPDU ()
```

12.2.4 Member Function Documentation

12.2.4.1 AddPresentationContextAC()

```
void gdcm::network::AAssociateACPDU::AddPresentationContextAC (  
    PresentationContextAC const & pcac)
```

12.2.4.2 GetNumberOfPresentationContextAC()

```
SizeType gdcm::network::AAssociateACPDU::GetNumberOfPresentationContextAC () const    [inline]
```

12.2.4.3 GetPresentationContextAC()

```
const PresentationContextAC & gdcm::network::AAssociateACPDU::GetPresentationContextAC (  
    SizeType i)    [inline]
```

References [gdcm_assert](#).

12.2.4.4 GetUserInfoInformation()

```
const UserInfoInformation & gdcm::network::AAssociateACPDU::GetUserInfoInformation () const    [inline]
```

12.2.4.5 InitFromRQ()

```
void gdcm::network::AAssociateACPDU::InitFromRQ (  
    AAssociateRQPDU const & rqpdu)
```

References [AAssociateRQPDU](#).

12.2.4.6 IsLastFragment()

```
bool gdcm::network::AAssociateACPDU::IsLastFragment () const    [inline], [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.2.4.7 Print()

```
void gdcm::network::AAssociateACPDU::Print (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.2.4.8 Read()

```
std::istream & gdcm::network::AAssociateACPDU::Read (  
    std::istream & is)    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.2.4.9 SetCalledAETitle()

```
void gdcm::network::AAssociateACPDU::SetCalledAETitle (  
    const char calledaetitle[16])    [protected]
```

12.2.4.10 SetCallingAETitle()

```
void gdcm::network::AAssociateACPDU::SetCallingAETitle (  
    const char callingaetitle[16])    [protected]
```

References [AAssociateRQPDU](#).

12.2.4.11 Size()

```
SizeType gdcm::network::AAssociateACPDU::Size () const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.2.4.12 Write()

```
const std::ostream & gdcm::network::AAssociateACPDU::Write (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.2.5 Friends And Related Symbol Documentation

12.2.5.1 AAssociateRQPDU

```
friend class AAssociateRQPDU    [friend]
```

References [AAssociateRQPDU](#).

Referenced by [AAssociateRQPDU](#), [InitFromRQ\(\)](#), and [SetCallingAETitle\(\)](#).

The documentation for this class was generated from the following file:

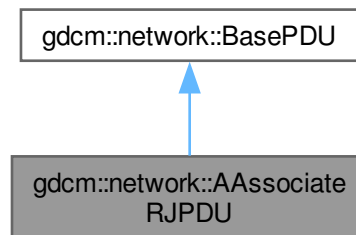
- [gdcmAAssociateACPDU.h](#)

12.3 gdcm::network::AAssociateRJPDU Class Reference

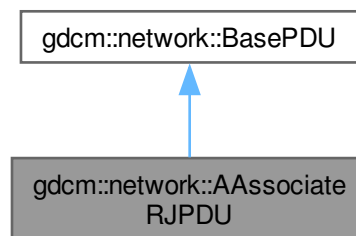
[AAssociateRJPDU](#).

```
#include <gdcmAAssociateRJPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

- [AAssociateRJPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcmm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

12.3.1 Detailed Description

[AAssociateRJPDU](#).

[Table 9-21](#) ASSOCIATE-RJ PDU FIELDS

12.3.2 Constructor & Destructor Documentation

12.3.2.1 AAssociateRJPDU()

`gdcmm::network::AAssociateRJPDU::AAssociateRJPDU ()`

12.3.3 Member Function Documentation

12.3.3.1 IsLastFragment()

`bool gdcmm::network::AAssociateRJPDU::IsLastFragment () const` [\[inline\]](#), [\[override\]](#), [\[virtual\]](#)

Implements [gdcmm::network::BasePDU](#).

12.3.3.2 Print()

`void gdcmm::network::AAssociateRJPDU::Print (std::ostream & os) const` [\[override\]](#), [\[virtual\]](#)

Implements [gdcmm::network::BasePDU](#).

12.3.3.3 Read()

`std::istream & gdcmm::network::AAssociateRJPDU::Read (std::istream & is)` [\[override\]](#), [\[virtual\]](#)

Implements [gdcmm::network::BasePDU](#).

12.3.3.4 Size()

`size_t gdcmm::network::AAssociateRJPDU::Size () const` [\[override\]](#), [\[virtual\]](#)

Implements [gdcmm::network::BasePDU](#).

12.3.3.5 Write()

```
const std::ostream & gdcm::network::AAssociateRJPDU::Write (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

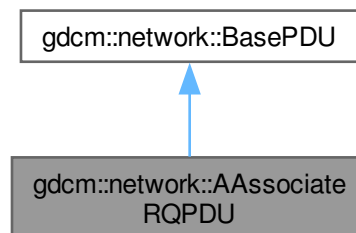
- [gdcmAAssociateRJPDU.h](#)

12.4 gdcm::network::AAssociateRQPDU Class Reference

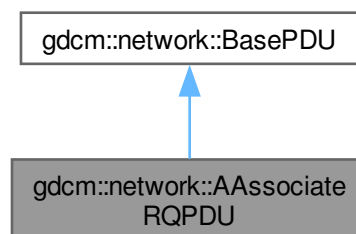
[AAssociateRQPDU](#).

```
#include <gdcmAAssociateRQPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRQPDU:



Collaboration diagram for gdcm::network::AAssociateRQPDU:



Public Types

- typedef std::vector< [PresentationContextRQ](#) > [PresentationContextArrayType](#)
- typedef std::vector< [PresentationContextRQ](#) >::size_type [SizeType](#)

Public Member Functions

- [AAAssociateRQPDU](#) ()
- [AAAssociateRQPDU](#) (const [AAAssociateRQPDU](#) &pdu)
- void [AddPresentationContext](#) ([PresentationContextRQ](#) const &pc)
- std::string [GetCalledAETitle](#) () const
- std::string [GetCallingAETitle](#) () const
- [SizeType](#) [GetNumberOfPresentationContext](#) () const
- [PresentationContextRQ](#) const & [GetPresentationContext](#) ([SizeType](#) i) const
- const [PresentationContextRQ](#) * [GetPresentationContextByAbstractSyntax](#) ([AbstractSyntax](#) const &absyn) const
- const [PresentationContextRQ](#) * [GetPresentationContextByID](#) (uint8_t i) const
- [PresentationContextArrayType](#) const & [GetPresentationContexts](#) ()
- const [UserInformation](#) & [GetUserInformation](#) () const
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- void [SetCalledAETitle](#) (const char calledaetitle[16])
Set the Called AE Title.
- void [SetCallingAETitle](#) (const char callingaetitle[16])
Set the Calling AE Title.
- void [SetUserInformation](#) ([UserInformation](#) const &ui)
- size_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcn::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

Static Public Member Functions

- static bool [IsAETitleValid](#) (const char title[16])
Check whether or not the.

Protected Member Functions

- std::string [GetReserved43_74](#) () const

Friends

- class [AAAssociateACPDU](#)

12.4.1 Detailed Description

[AAssociateRQPDU](#).

[Table 9-11](#) ASSOCIATE-RQ PDU fields

12.4.2 Member Typedef Documentation

12.4.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContextRQ> gdcmm::network::AAssociateRQPDU::PresentationContextArrayType
```

12.4.2.2 SizeType

```
typedef std::vector<PresentationContextRQ>::size_type gdcmm::network::AAssociateRQPDU::SizeType
```

12.4.3 Constructor & Destructor Documentation

12.4.3.1 AAssociateRQPDU() [1/2]

```
gdcmm::network::AAssociateRQPDU::AAssociateRQPDU \(\)
```

Referenced by [AAssociateRQPDU\(\)](#).

12.4.3.2 AAssociateRQPDU() [2/2]

```
gdcmm::network::AAssociateRQPDU::AAssociateRQPDU \(  

    const AAssociateRQPDU & pdu) \[inline\]
```

References [AAssociateRQPDU\(\)](#), and [gdcmm_assert](#).

12.4.4 Member Function Documentation

12.4.4.1 AddPresentationContext()

```
void gdcmm::network::AAssociateRQPDU::AddPresentationContext \(  

    PresentationContextRQ const & pc)
```

12.4.4.2 GetCalledAETitle()

```
std::string gdcmm::network::AAssociateRQPDU::GetCalledAETitle \(\) const \[inline\]
```

12.4.4.3 GetCallingAETitle()

std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle () const [inline]

12.4.4.4 GetNumberOfPresentationContext()

[SizeType](#) gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext () const [inline]

12.4.4.5 GetPresentationContext()

[PresentationContextRQ](#) const & gdcm::network::AAssociateRQPDU::GetPresentationContext (
[SizeType](#) i) const [inline]

References [gdcm_assert](#).

12.4.4.6 GetPresentationContextByAbstractSyntax()

const [PresentationContextRQ](#) * gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (
[AbstractSyntax](#) const & absyn) const

12.4.4.7 GetPresentationContextByID()

const [PresentationContextRQ](#) * gdcm::network::AAssociateRQPDU::GetPresentationContextByID (
uint8_t i) const

12.4.4.8 GetPresentationContexts()

[PresentationContextArrayType](#) const & gdcm::network::AAssociateRQPDU::GetPresentationContexts () [inline]

12.4.4.9 GetReserved43_74()

std::string gdcm::network::AAssociateRQPDU::GetReserved43_74 () const [protected]

12.4.4.10 GetUserInfoInformation()

const [UserInfoInformation](#) & gdcm::network::AAssociateRQPDU::GetUserInfoInformation () const [inline]

12.4.4.11 IsAETitleValid()

```
bool gdcn::network::AAAssociateRQPDU::IsAETitleValid (
    const char title[16]) [static]
```

Check whether or not the.

Parameters

title	is a valid AE title
-------	---------------------

12.4.4.12 IsLastFragment()

```
bool gdcn::network::AAAssociateRQPDU::IsLastFragment () const [inline], [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

12.4.4.13 Print()

```
void gdcn::network::AAAssociateRQPDU::Print (
    std::ostream & os) const [override], [virtual]
```

This function will initialize an [AAAssociateACPDU](#) from the fields in the [AAAssociateRQPDU](#) structure

Implements [gdcn::network::BasePDU](#).

12.4.4.14 Read()

```
std::istream & gdcn::network::AAAssociateRQPDU::Read (
    std::istream & is) [override], [virtual]
```

Implements [gdcn::network::BasePDU](#).

12.4.4.15 SetCalledAETitle()

```
void gdcn::network::AAAssociateRQPDU::SetCalledAETitle (
    const char calledaetitle[16])
```

Set the Called AE Title.

12.4.4.16 SetCallingAETitle()

```
void gdcn::network::AAAssociateRQPDU::SetCallingAETitle (
    const char callingaetitle[16])
```

Set the Calling AE Title.

12.4.4.17 SetUserInfoInformation()

```
void gdcm::network::AAssociateRQPDU::SetUserInfoInformation (  
    UserInfoInformation const & ui)
```

12.4.4.18 Size()

```
size_t gdcm::network::AAssociateRQPDU::Size () const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.4.4.19 Write()

```
const std::ostream & gdcm::network::AAssociateRQPDU::Write (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.4.5 Friends And Related Symbol Documentation

12.4.5.1 AAssociateACPDU

```
friend class AAssociateACPDU    [friend]
```

References [AAssociateACPDU](#).

Referenced by [AAssociateACPDU](#).

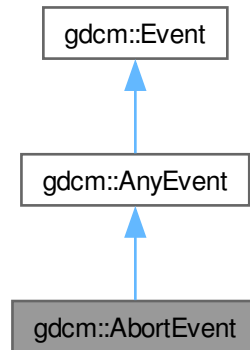
The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

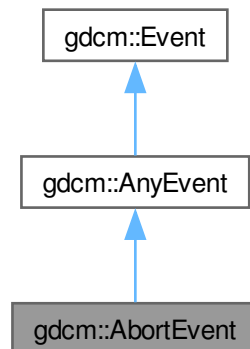
12.5 gdcm::AbortEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for `gdcm::AbortEvent`:



Collaboration diagram for `gdcm::AbortEvent`:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()

- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.6 gdcm::network::AbstractSyntax Class Reference

[AbstractSyntax](#).

```
#include <gdcmAbstractSyntax.h>
```

Public Member Functions

- [AbstractSyntax](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- const char * [GetName](#) () const
- bool [operator==](#) (const [AbstractSyntax](#) &as) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char *name)
- void [SetNameFromUID](#) ([UIDs::TSName](#) tsname)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.6.1 Detailed Description

[AbstractSyntax](#).

[Table 9-14](#) ABSTRACT SYNTAX SUB-ITEM FIELDS

12.6.2 Constructor & Destructor Documentation

12.6.2.1 AbstractSyntax()

gdcm::network::AbstractSyntax::AbstractSyntax ()

Referenced by [operator==\(\)](#).

12.6.3 Member Function Documentation

12.6.3.1 GetAsDataElement()

[DataElement](#) gdcm::network::AbstractSyntax::GetAsDataElement () const

12.6.3.2 GetName()

const char * gdcm::network::AbstractSyntax::GetName () const [inline]

12.6.3.3 operator==()

bool gdcm::network::AbstractSyntax::operator== (
const [AbstractSyntax](#) & as) const [inline]

References [AbstractSyntax\(\)](#).

12.6.3.4 Print()

void gdcm::network::AbstractSyntax::Print (
std::ostream & os) const

12.6.3.5 Read()

std::istream & gdcm::network::AbstractSyntax::Read (
std::istream & is)

12.6.3.6 SetName()

void gdcm::network::AbstractSyntax::SetName (
const char * name) [inline]

12.6.3.7 SetNameFromUID()

void gdcm::network::AbstractSyntax::SetNameFromUID (
[UIDs::TSName](#) tsname)

12.6.3.8 Size()

size_t gdcm::network::AbstractSyntax::Size () const

12.6.3.9 Write()

```
const std::ostream & gdcM::network::AbstractSyntax::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

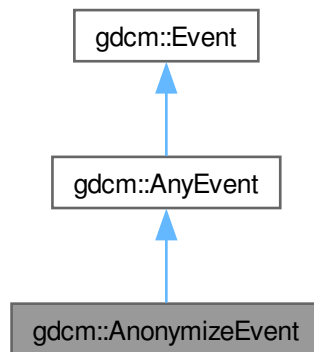
- [gdcMAbstractSyntax.h](#)

12.7 gdcM::AnonymizeEvent Class Reference

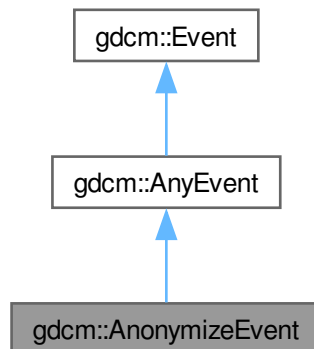
[AnonymizeEvent](#).

```
#include <gdcMAnonymizeEvent.h>
```

Inheritance diagram for gdcM::AnonymizeEvent:



Collaboration diagram for `gdcm::AnonymizeEvent`:



Public Types

- typedef [AnonymizeEvent](#) [Self](#)
- typedef [AnyEvent](#) [Superclass](#)

Public Member Functions

- [AnonymizeEvent](#) (const [Self](#) &s)
- [AnonymizeEvent](#) ([Tag](#) const &tag=0)
- [~AnonymizeEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const override
- const char * [GetEventName](#) () const override
- [Tag](#) const & [GetTag](#) () const
- [::gdcm::Event](#) * [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetTag](#) (const [Tag](#) &t)

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

12.7.1 Detailed Description

[AnonymizeEvent](#).

Special type of event triggered during the Anonymization process

See also

[Anonymizer](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.7.2 Member Typedef Documentation

12.7.2.1 Self

typedef [AnonymizeEvent](#) gdcm::AnonymizeEvent::Self

12.7.2.2 Superclass

typedef [AnyEvent](#) gdcm::AnonymizeEvent::Superclass

12.7.3 Constructor & Destructor Documentation

12.7.3.1 AnonymizeEvent() [1/2]

gdcm::AnonymizeEvent::AnonymizeEvent (
 [Tag](#) const & tag = 0) [inline]

12.7.3.2 ~AnonymizeEvent()

gdcm::AnonymizeEvent::~~AnonymizeEvent () [override], [default]

12.7.3.3 AnonymizeEvent() [2/2]

gdcm::AnonymizeEvent::AnonymizeEvent (
 const [Self](#) & s) [inline]

12.7.4 Member Function Documentation

12.7.4.1 CheckEvent()

```
bool gdcmm::AnonymizeEvent::CheckEvent (
    const ::gdcmm::Event * e) const    [inline], [override]
```

12.7.4.2 GetEventName()

```
const char * gdcmm::AnonymizeEvent::GetEventName () const    [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

12.7.4.3 GetTag()

```
Tag const & gdcmm::AnonymizeEvent::GetTag () const    [inline]
```

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.7.4.4 MakeObject()

```
::gdcmm::Event * gdcmm::AnonymizeEvent::MakeObject () const    [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcmm::Event](#).

12.7.4.5 operator=()

```
void gdcmm::AnonymizeEvent::operator= (
    const Self & )    [delete]
```

12.7.4.6 SetTag()

```
void gdcmm::AnonymizeEvent::SetTag (
    const Tag & t)    [inline]
```

The documentation for this class was generated from the following file:

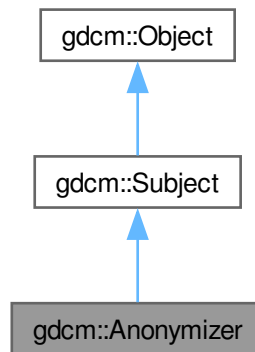
- [gdcmmAnonymizeEvent.h](#)

12.8 gdcm::Anonymizer Class Reference

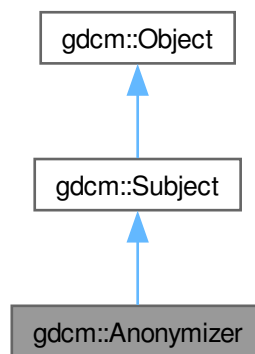
[Anonymizer.](#)

```
#include <gdcmAnonymizer.h>
```

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- [Anonymizer](#) ()
- [~Anonymizer](#) () override
- bool [BasicApplicationLevelConfidentialityProfile](#) (bool deidentify=true)
- bool [Clear](#) ([PrivateTag](#) const &pt)
- bool [Clear](#) ([Tag](#) const &t)
 - Identical to 'Empty' except no action is done when tag is not present.
- bool [Empty](#) ([PrivateTag](#) const &pt)
- bool [Empty](#) ([Tag](#) const &t)
 - Make [Tag](#) t empty (if not found tag will be created).
- const [CryptographicMessageSyntax](#) * [GetCryptographicMessageSyntax](#) () const
- [File](#) & [GetFile](#) ()
- bool [Remove](#) ([PrivateTag](#) const &pt)
- bool [Remove](#) ([Tag](#) const &t)
 - remove a tag (even a SQ can be removed)
- bool [RemoveGroupLength](#) ()
 - Main function that loop over all elements and remove group length.
- bool [RemovePrivateTags](#) ()
 - Main function that loop over all elements and remove private tags.
- bool [RemoveRetired](#) ()
 - Main function that loop over all elements and remove retired element.
- bool [Replace](#) ([PrivateTag](#) const &t, const char *value)
- bool [Replace](#) ([PrivateTag](#) const &t, const char *value, [VL](#) const &vl)
- bool [Replace](#) ([Tag](#) const &t, const char *value)
- bool [Replace](#) ([Tag](#) const &t, const char *value, [VL](#) const &vl)
- void [SetCryptographicMessageSyntax](#) ([CryptographicMessageSyntax](#) *cms)
 - Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.
- void [SetFile](#) (const [File](#) &f)
 - Set/Get [File](#).

Public Member Functions inherited from [gdcms::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static void [ClearInternalUIDs](#) ()
- static std::vector< [Tag](#) > [GetBasicApplicationLevelConfidentialityProfileAttributes](#) ()
- Return the list of [Tag](#) that will be considered when anonymizing a DICOM file.
- static [SmartPointer](#)< [Anonymizer](#) > [New](#) ()
- for wrapped language: instantiate a reference counted object

Protected Member Functions

- bool [BALCPPProtect](#) ([DataSet](#) &ds, [Tag](#) const &tag, const [IOD](#) &iod)
- bool [CanEmptyTag](#) ([Tag](#) const &tag, const [IOD](#) &iod) const
- void [RecurseDataSet](#) ([DataSet](#) &ds)

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.8.1 Detailed Description

[Anonymizer](#).

This class is a multi purpose anonymizer. It can work in 2 mode:

- Full (irreversible) anonymizer (aka dumb mode)
- reversible de-identifier/re-identifier (aka smart mode). This implements the Basic Application Level Confidentiality Profile, DICOM PS 3.15-2009

1. dumb mode This is a dumb anonymizer implementation. All it allows user is simple operation such as:

[Tag](#) based functions:

- complete removal of DICOM attribute (Remove)
- make a tag empty, ie make it's length 0 (Empty)

- replace with another string-based value (Replace)

[DataSet](#) based functions:

- Remove all group length attribute from a DICOM dataset (Group Length element are deprecated, DICOM 2008)
- Remove all private attributes
- Remove all retired attributes

All function calls actually execute the user specified request. Previous implementation were calling a general Anonymize function but traversing a `std::set` is $O(n)$ operation, while a simple user specified request is $O(\log(n))$ operation. So 'm' user interaction is $O(m \cdot \log(n))$ which is $< O(n)$ complexity.

1. smart mode this mode implements the Basic Application Level Confidentiality Profile (DICOM PS 3.15-2008) In this case, it is extremely important to use the same [Anonymizer](#) class when anonymizing a [FileSet](#). Once the [Anonymizer](#) is destroyed its memory of known (already processed) [UIDs](#) will be lost. which will make the anonymizer behaves incorrectly for attributes such as [Series](#) [UID](#) [Study](#) [UID](#) where user want some consistency. When attribute is [Type](#) 1 / [Type](#) 1C, a dummy generator will take in the existing value and produce a dummy value (a sha1 representation). sha1 algorithm is considered to be cryptographically strong (compared to md5sum) so that we meet the following two conditions:
 - Produce the same dummy value for the same input value
 - do not provide an easy way to retrieve the original value from the sha1 generated value

This class implement the Subject/Observer pattern trigger the following event:

- [AnonymizeEvent](#)
- [IterationEvent](#)
- [StartEvent](#)
- [EndEvent](#)

See also

[CryptographicMessageSyntax](#)

Examples

[ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

12.8.2 Constructor & Destructor Documentation

12.8.2.1 Anonymizer()

`gdcm::Anonymizer::Anonymizer ()` [inline]

Referenced by [New\(\)](#).

12.8.2.2 ~Anonymizer()

gdcm::Anonymizer::~~Anonymizer () [override]

12.8.3 Member Function Documentation

12.8.3.1 BALCPPProtect()

```
bool gdcm::Anonymizer::BALCPPProtect (  
    DataSet & ds,  
    Tag const & tag,  
    const IOD & iod) [protected]
```

12.8.3.2 BasicApplicationLevelConfidentialityProfile()

```
bool gdcm::Anonymizer::BasicApplicationLevelConfidentialityProfile (  
    bool deidentify = true)
```

PS 3.15 / E.1.1 De-Identifier An Application may claim conformance to the Basic Application Level Confidentiality Profile as a deidentifier if it protects all Attributes that might be used by unauthorized entities to identify the patient. NOT THREAD SAFE

Examples

[BasicAnonymizer.cs](#).

12.8.3.3 CanEmptyTag()

```
bool gdcm::Anonymizer::CanEmptyTag (  
    Tag const & tag,  
    const IOD & iod) const [protected]
```

12.8.3.4 Clear() [1/2]

```
bool gdcm::Anonymizer::Clear (  
    PrivateTag const & pt)
```

12.8.3.5 Clear() [2/2]

```
bool gdcm::Anonymizer::Clear (  
    Tag const & t)
```

Identical to 'Empty' except no action is done when tag is not present.

12.8.3.6 ClearInternalUIDs()

```
void gdcmm::Anonymizer::ClearInternalUIDs () [static]
```

Clear the internal mapping of real [UIDs](#) to generated [UIDs](#)

Warning

the mapping is definitely lost

12.8.3.7 Empty() [1/2]

```
bool gdcmm::Anonymizer::Empty (
    PrivateTag const & pt)
```

Make [PrivateTag](#) pt empty (if not found tag will be created) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace.

12.8.3.8 Empty() [2/2]

```
bool gdcmm::Anonymizer::Empty (
    Tag const & t)
```

Make [Tag](#) t empty (if not found tag will be created).

Examples

[CreateJPIPDataSet.cxx](#).

12.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
std::vector< Tag > gdcmm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes () [static]
```

Return the list of [Tag](#) that will be considered when anonymizing a DICOM file.

Examples

[GenFakeIdentifyFile.cxx](#), and [TraverseModules.cxx](#).

12.8.3.10 GetCryptographicMessageSyntax()

```
const CryptographicMessageSyntax * gdcmm::Anonymizer::GetCryptographicMessageSyntax () const
```

12.8.3.11 GetFile()

[File](#) & gdcm::Anonymizer::GetFile () [inline]

Examples

[BasicAnonymizer.cs](#), and [ManipulateFile.cs](#).

12.8.3.12 New()

[SmartPointer](#)< [Anonymizer](#) > gdcm::Anonymizer::New () [inline], [static]

for wrapped language: instantiate a reference counted object

References [Anonymizer\(\)](#).

12.8.3.13 RecurseDataSet()

void gdcm::Anonymizer::RecurseDataSet (
 [DataSet](#) & ds) [protected]

12.8.3.14 Remove() [1/2]

bool gdcm::Anonymizer::Remove (
 [PrivateTag](#) const & pt)

remove a private tag (even a SQ can be removed) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace. When the private reservation becomes empty, no check is done to automatically remove the private creator

12.8.3.15 Remove() [2/2]

bool gdcm::Anonymizer::Remove (
 [Tag](#) const & t)

remove a tag (even a SQ can be removed)

12.8.3.16 RemoveGroupLength()

bool gdcm::Anonymizer::RemoveGroupLength ()

Main function that loop over all elements and remove group length.

Examples

[ClinicalTrialAnnotate.cxx](#), and [ManipulateFile.cs](#).

12.8.3.17 RemovePrivateTags()

```
bool gdcm::Anonymizer::RemovePrivateTags ()
```

Main function that loop over all elements and remove private tags.

Examples

[ClinicalTrialAnnotate.cxx](#), and [ManipulateFile.cs](#).

12.8.3.18 RemoveRetired()

```
bool gdcm::Anonymizer::RemoveRetired ()
```

Main function that loop over all elements and remove retired element.

12.8.3.19 Replace() [1/4]

```
bool gdcm::Anonymizer::Replace (  
    PrivateTag const & t,  
    const char * value)
```

12.8.3.20 Replace() [2/4]

```
bool gdcm::Anonymizer::Replace (  
    PrivateTag const & t,  
    const char * value,  
    VL const & vl)
```

12.8.3.21 Replace() [3/4]

```
bool gdcm::Anonymizer::Replace (  
    Tag const & t,  
    const char * value)
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII

Examples

[ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

12.8.3.22 Replace() [4/4]

```
bool gdcm::Anonymizer::Replace (  
    Tag const & t,  
    const char * value,  
    VL const & vl)
```

when the value contains `\0`, it is a good idea to specify the length. This function is required when dealing with `VRBINARY` tag

12.8.3.23 SetCryptographicMessageSyntax()

```
void gdcm::Anonymizer::SetCryptographicMessageSyntax (  
    CryptographicMessageSyntax * cms)
```

Set/Get CMS key that will be used to encrypt the dataset within `BasicApplicationLevelConfidentialityProfile`.

Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.8.3.24 SetFile()

```
void gdcm::Anonymizer::SetFile (  
    const File & f) [inline]
```

Set/Get `File`.

Examples

[BasicAnonymizer.cs](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [ManipulateFile.cs](#), and [MpegVideoInfo.cs](#).

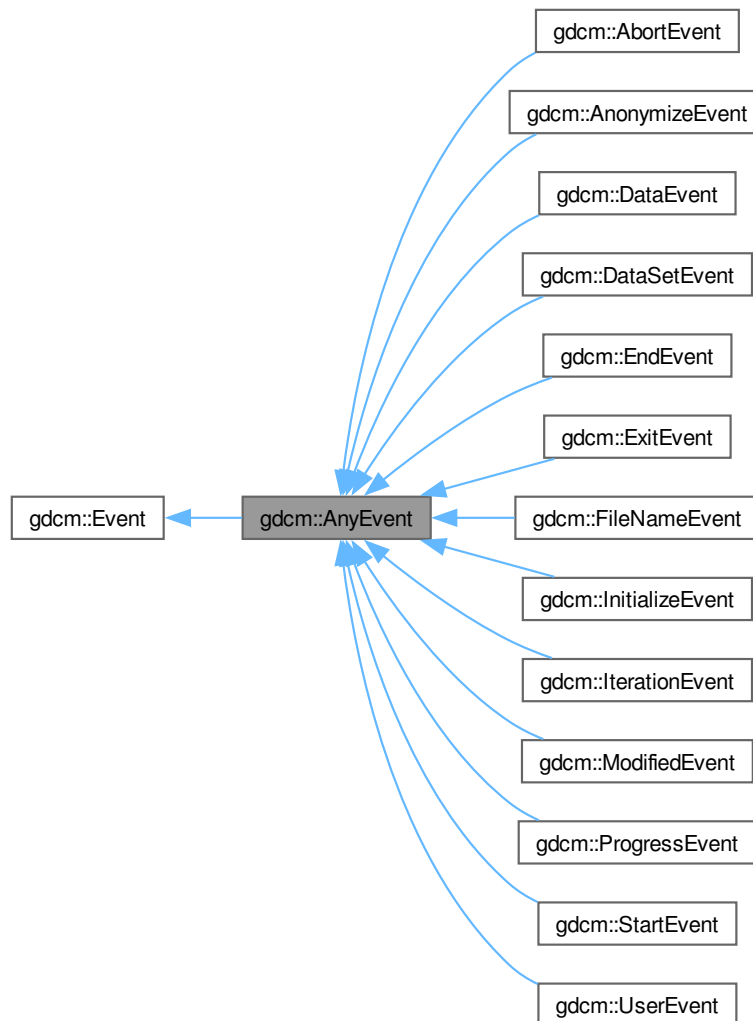
The documentation for this class was generated from the following file:

- [gdcmAnonymizer.h](#)

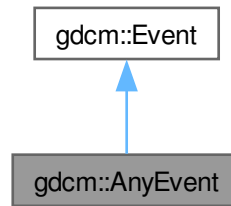
12.9 gdcm::AnyEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.10 gdcm::network::ApplicationContext Class Reference

[ApplicationContext](#).

```
#include <gdcmApplicationContext.h>
```

Public Member Functions

- [ApplicationContext](#) ()
- const char * [GetName](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char *name)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.10.1 Detailed Description

[ApplicationContext](#).

[Table](#) 9-12 APPLICATION CONTEXT ITEM FIELDS

[Todo](#) Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

12.10.2 Constructor & Destructor Documentation

12.10.2.1 ApplicationContext()

```
gdcmm::network::ApplicationContext::ApplicationContext ()
```

12.10.3 Member Function Documentation

12.10.3.1 GetName()

```
const char * gdcmm::network::ApplicationContext::GetName () const [inline]
```

12.10.3.2 Print()

```
void gdcmm::network::ApplicationContext::Print (  
    std::ostream & os) const
```

12.10.3.3 Read()

```
std::istream & gdcmm::network::ApplicationContext::Read (  
    std::istream & is)
```

12.10.3.4 SetName()

```
void gdcmm::network::ApplicationContext::SetName (  
    const char * name) [inline]
```

12.10.3.5 Size()

```
size_t gdcmm::network::ApplicationContext::Size () const
```


12.10.3.6 Write()

```
const std::ostream & gdcm::network::ApplicationContext::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

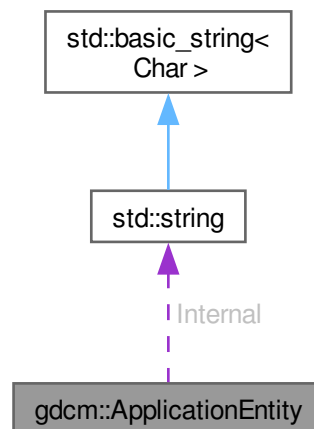
- [gdcmApplicationContext.h](#)

12.11 gdcm::ApplicationEntity Class Reference

[ApplicationEntity](#).

```
#include <gdcmApplicationEntity.h>
```

Collaboration diagram for gdcm::ApplicationEntity:



Public Member Functions

- bool [IsValid](#) () const
- void [Print](#) (std::ostream &os) const
- void [SetBlob](#) (const std::vector< char > &v)
- void [Squeeze](#) ()

Public Attributes

- std::string [Internal](#)

Static Public Attributes

- static const unsigned int [MaxLength](#) = 16
- static const unsigned int [MaxNumberOfComponents](#) = 1
- static const char [Padding](#) = ' '
- static const char [Separator](#) = ' '

12.11.1 Detailed Description

[ApplicationEntity](#).

- AE Application Entity
- A string of characters that identifies an Application Entity with leading and trailing spaces (20H) being non-significant. A value consisting solely of spaces shall not be used.
- Default Character Repertoire excluding character code 5CH (the BACKSLASH \ in ISO-IR 6), and control characters LF, FF, CR and ESC.
- 16 bytes maximum

12.11.2 Member Function Documentation

12.11.2.1 IsValid()

```
bool gdcmm::ApplicationEntity::IsValid () const [inline]
```

12.11.2.2 Print()

```
void gdcmm::ApplicationEntity::Print (  
    std::ostream & os) const [inline]
```

References [gdcmm_assert](#).

12.11.2.3 SetBlob()

```
void gdcmm::ApplicationEntity::SetBlob (  
    const std::vector< char > & v) [inline]
```

References [gdcmm_assert](#).

12.11.2.4 Squeeze()

```
void gdcmm::ApplicationEntity::Squeeze () [inline]
```

12.11.3 Member Data Documentation

12.11.3.1 Internal

`std::string gdcmm::ApplicationEntity::Internal`

12.11.3.2 MaxLength

`const unsigned int gdcmm::ApplicationEntity::MaxLength = 16` [static]

12.11.3.3 MaxNumberOfComponents

`const unsigned int gdcmm::ApplicationEntity::MaxNumberOfComponents = 1` [static]

12.11.3.4 Padding

`const char gdcmm::ApplicationEntity::Padding = ' '` [static]

12.11.3.5 Separator

`const char gdcmm::ApplicationEntity::Separator = ' '` [static]

The documentation for this class was generated from the following file:

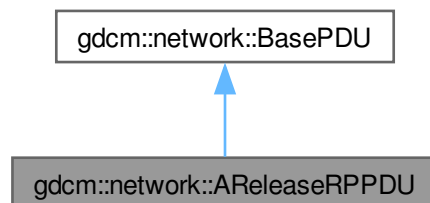
- [gdcmmApplicationEntity.h](#)

12.12 gdcmm::network::AReleaseRPPDU Class Reference

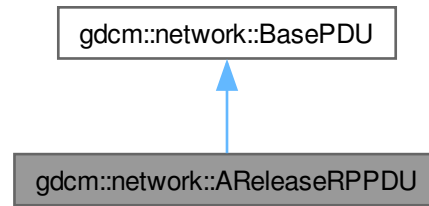
[AReleaseRPPDU](#).

```
#include <gdcmmAReleaseRPPDU.h>
```

Inheritance diagram for gdcmm::network::AReleaseRPPDU:



Collaboration diagram for `gdcm::network::AReleaseRPPDU`:



Public Member Functions

- [AReleaseRPPDU](#) ()
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcm::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

12.12.1 Detailed Description

[AReleaseRPPDU](#).

[Table 9-25](#) A-RELEASE-RP PDU fields

12.12.2 Constructor & Destructor Documentation

12.12.2.1 AReleaseRPPDU()

`gdcm::network::AReleaseRPPDU::AReleaseRPPDU ()`

12.12.3 Member Function Documentation

12.12.3.1 IsLastFragment()

bool `gdcm::network::AReleaseRPPDU::IsLastFragment () const` [inline], [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.12.3.2 Print()

```
void gdcm::network::AReleaseRPPDU::Print (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.12.3.3 Read()

```
std::istream & gdcm::network::AReleaseRPPDU::Read (  
    std::istream & is)    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.12.3.4 Size()

```
size_t gdcm::network::AReleaseRPPDU::Size () const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

12.12.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRPPDU::Write (  
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

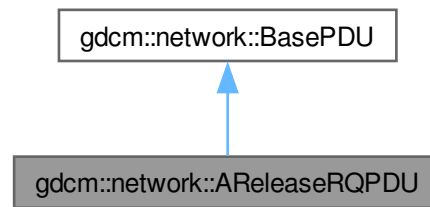
- [gdcmAReleaseRPPDU.h](#)

12.13 gdcm::network::AReleaseRQPDU Class Reference

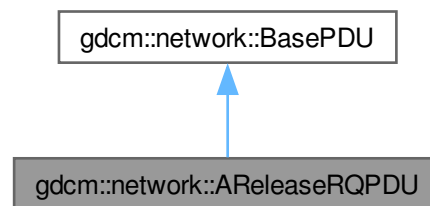
[AReleaseRQPDU](#).

```
#include <gdcmAReleaseRQPDU.h>
```

Inheritance diagram for `gdcm::network::AReleaseRQPDU`:



Collaboration diagram for `gdcm::network::AReleaseRQPDU`:



Public Member Functions

- [AReleaseRQPDU](#) ()
- `bool` [IsLastFragment](#) () const override
- `void` [Print](#) (std::ostream &os) const override
- `std::istream &` [Read](#) (std::istream &is) override
- `size_t` [Size](#) () const override
- `const std::ostream &` [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcm::network::BasePDU](#)

- `virtual` [~BasePDU](#) ()=default

12.13.1 Detailed Description

[AReleaseRQPDU](#).

[Table 9-24](#) A-RELEASE-RQ PDU FIELDS

12.13.2 Constructor & Destructor Documentation

12.13.2.1 AReleaseRQPDU()

gdcm::network::AReleaseRQPDU::AReleaseRQPDU ()

12.13.3 Member Function Documentation

12.13.3.1 IsLastFragment()

bool gdcm::network::AReleaseRQPDU::IsLastFragment () const [inline], [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.13.3.2 Print()

void gdcm::network::AReleaseRQPDU::Print (
std::ostream & os) const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.13.3.3 Read()

std::istream & gdcm::network::AReleaseRQPDU::Read (
std::istream & is) [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.13.3.4 Size()

size_t gdcm::network::AReleaseRQPDU::Size () const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.13.3.5 Write()

const std::ostream & gdcm::network::AReleaseRQPDU::Write (
std::ostream & os) const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

- [gdcmAReleaseRQPDU.h](#)

12.14 gdcm::network::ARTIMTimer Class Reference

[ARTIMTimer](#).

```
#include <gdcmARTIMTimer.h>
```

Public Member Functions

- [ARTIMTimer](#) ()
- double [GetElapsedTime](#) () const
- bool [GetHasExpired](#) () const
- double [GetTimeout](#) () const
- void [SetTimeout](#) (double inTimeout)
- void [Start](#) ()
- void [Stop](#) ()

12.14.1 Detailed Description

[ARTIMTimer](#).

This file contains the code for the ARTIM timer.

Basically, the ARTIM timer will just get the wall time when it's started, and then can be queried for the current time, and then can be stopped (ie, the start time reset).

Because we're trying to do this without threading, we should be able to 'start' the ARTIM timer by this mechanism, and then when waiting for a particular response, tight loop that with sleep calls and determinations of when the ARTIM timer has reached its peak. As such, this isn't a strict 'timer' in the traditional sense of the word, but more of a time keeper.

There can be only one ARTIM timer per connection.

12.14.2 Constructor & Destructor Documentation

12.14.2.1 ARTIMTimer()

```
gdcm::network::ARTIMTimer::ARTIMTimer ()
```

12.14.3 Member Function Documentation

12.14.3.1 GetElapsedTime()

```
double gdcm::network::ARTIMTimer::GetElapsedTime () const
```


12.14.3.2 GetHasExpired()

```
bool gdcm::network::ARTIMTimer::GetHasExpired () const
```

12.14.3.3 GetTimeout()

```
double gdcm::network::ARTIMTimer::GetTimeout () const
```

12.14.3.4 SetTimeout()

```
void gdcm::network::ARTIMTimer::SetTimeout (  
    double inTimeout)
```

12.14.3.5 Start()

```
void gdcm::network::ARTIMTimer::Start ()
```

12.14.3.6 Stop()

```
void gdcm::network::ARTIMTimer::Stop ()
```

The documentation for this class was generated from the following file:

- [gdcmARTIMTimer.h](#)

12.15 gdcm::ASN1 Class Reference

Class for [ASN1](#).

```
#include <gdcmASN1.h>
```

Public Member Functions

- [ASN1](#) ()
- [ASN1](#) (const [ASN1](#) &)=delete
- [~ASN1](#) ()
- void [operator=](#) (const [ASN1](#) &)=delete

Static Public Member Functions

- static bool [ParseDump](#) (const char *array, size_t length)
- static bool [ParseDumpFile](#) (const char *filename)

Protected Member Functions

- int [TestPBKDF2](#) ()

12.15.1 Detailed Description

Class for [ASN1](#).

12.15.2 Constructor & Destructor Documentation

12.15.2.1 [ASN1](#)() [1/2]

gdcmm::ASN1::ASN1 ()

Referenced by [ASN1\(\)](#), and [operator=\(\)](#).

12.15.2.2 [~ASN1](#)()

gdcmm::ASN1::~~ASN1 ()

12.15.2.3 [ASN1](#)() [2/2]

gdcmm::ASN1::ASN1 (
 const ASN1 &) [delete]

References [ASN1\(\)](#).

12.15.3 Member Function Documentation

12.15.3.1 [operator=](#)()

void gdcmm::ASN1::operator= (
 const [ASN1](#) &) [delete]

References [ASN1\(\)](#).

12.15.3.2 [ParseDump](#)()

bool gdcmm::ASN1::ParseDump (
 const char * array,
 size_t length) [static]

12.15.3.3 ParseDumpFile()

```
bool gdcm::ASN1::ParseDumpFile (
    const char * filename) [static]
```

12.15.3.4 TestPBKDF2()

```
int gdcm::ASN1::TestPBKDF2 () [protected]
```

The documentation for this class was generated from the following file:

- [gdcmASN1.h](#)

12.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

[AsynchronousOperationsWindowSub.](#)

```
#include <gdcmAsynchronousOperationsWindowSub.h>
```

Public Member Functions

- [AsynchronousOperationsWindowSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.16.1 Detailed Description

[AsynchronousOperationsWindowSub.](#)

PS 3.7 [Table](#) D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-↔ RQ)

12.16.2 Constructor & Destructor Documentation

12.16.2.1 AsynchronousOperationsWindowSub()

```
gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ()
```

12.16.3 Member Function Documentation

12.16.3.1 Print()

```
void gdcmm::network::AsynchronousOperationsWindowSub::Print (
    std::ostream & os) const
```

12.16.3.2 Read()

```
std::istream & gdcmm::network::AsynchronousOperationsWindowSub::Read (
    std::istream & is)
```

12.16.3.3 Size()

```
size_t gdcmm::network::AsynchronousOperationsWindowSub::Size () const
```

12.16.3.4 Write()

```
const std::ostream & gdcmm::network::AsynchronousOperationsWindowSub::Write (
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

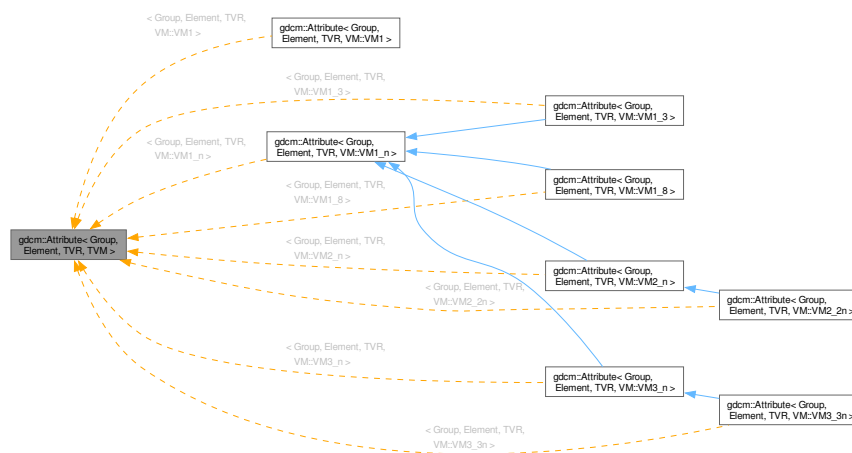
- [gdcmmAsynchronousOperationsWindowSub.h](#)

12.17 gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference

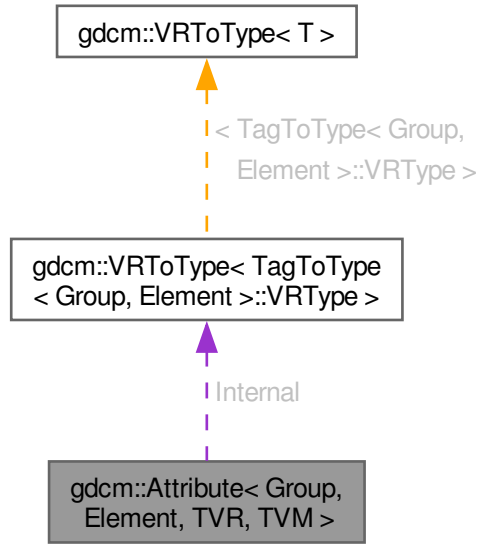
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

```
#include <gdcmmAttribute.h>
```

Inheritance diagram for gdcmm::Attribute< Group, Element, TVR, TVM >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



Public Types

- enum { `VMType` = `VMToLength<TVM>::Length` }
- typedef `VRToType< TVR >::Type` `ArrayType`

Public Member Functions

- `GDCM_STATIC_ASSERT` (((((`VR::VRType`) `TVR` &`VR::VR_VM1`) &&((`VM::VMType`) `TVM`==`VM::VM1`))||((`VR` &`VR::VR_VM1`)))
- `GDCM_STATIC_ASSERT` (((`VM::VMType`) `TVM` &(`VM::VMType`)(`TagToType< Group, Element >::VMType`)))
- `GDCM_STATIC_ASSERT` (((`VR::VRType`) `TVR` &(`VR::VRType`)(`TagToType< Group, Element >::VRType`)))
- `DataElement` `GetAsDataElement` () const
- unsigned int `GetNumberOfValues` () const
- `ArrayType` & `GetValue` (unsigned int idx=0)
- `ArrayType` const & `GetValue` (unsigned int idx=0) const
- const `ArrayType` * `GetValues` () const
- bool `operator!=` (const `Attribute` &att) const
- bool `operator<` (const `Attribute` &att) const
- bool `operator==` (const `Attribute` &att) const
- `ArrayType` & `operator[]` (unsigned int idx)
- `ArrayType` const & `operator[]` (unsigned int idx) const

- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [ArrayType](#) [Internal](#) [[VMToLength](#)< [TVM](#) >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.17.1 Detailed Description

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType,
int TVM = TagToType<Group, Element>::VMType>
class gdcmm::Attribute< Group, Element, TVR, TVM >
```

[Attribute](#) class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

Typical example that compile is: [Attribute<0x0008,0x9007>](#) a = {"ORIGINAL","PRIMARY","T1","↵NONE"};

Examples that will NOT compile are:

```
Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {}; // not enough parameters Attribute<0x0018,0x1182, VR::IS, VM::V
fd2 = {0,1,2}; // too many initializers Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2}; //
VM3 is not valid Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1}; // UL is not valid VR
```

Examples

[CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [Extracting_All_Resolution.cxx](#),
[Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#),
[GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#),
[ReadAndPrintAttributes.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [VolumeSorter.cxx](#),
[gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.17.2 Member Typedef Documentation

12.17.2.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

Examples

[ReadAndPrintAttributes.cxx](#).

12.17.3 Member Enumeration Documentation

12.17.3.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
anonymous enum
```

Enumerator

VMType	
--------	--

12.17.4 Member Function Documentation

12.17.4.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
gdcmm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||((VR::VRType) TVR
&VR::VR_VM1)) )
```

References [gdcmm::VM::VM1](#), and [gdcmm::VR::VR_VM1](#).

12.17.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
gdcmm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

12.17.4.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
gdcml::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.17.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
DataElement gdcml::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const [inline]
```

Examples

[CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), and [StreamImageReaderTest.cxx](#).

References [gdcml_assert](#), [GetNumberOfValues\(\)](#), [GetTag\(\)](#), [GetVR\(\)](#), [gdcml::DataElement::GetVR\(\)](#), [Internal](#), [gdcml::DataElement::SetByteValue\(\)](#), [gdcml::DataElement::SetVR\(\)](#), [gdcml::VR::SQ](#), [gdcml::VR::UI](#), and [gdcml::VR::VRASCII](#).

12.17.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
VM gdcml::Attribute< Group, Element, TVR, TVM >::GetDictVM () [inline], [static]
```

12.17.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
VR gdcml::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```

12.17.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
unsigned int gdcml::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const [inline]
```

Examples

[LargeVRDSExplicit.cxx](#).

Referenced by [GetAsDataElement\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [gdcml::Attribute< Group, Element, operator==\(\)](#), [Print\(\)](#), [SetByteValue\(\)](#), [SetByteValueNoSwap\(\)](#), [SetValue\(\)](#), and [SetValues\(\)](#).

12.17.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
```

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline], [static]
```

Examples

[PatchFile.cxx](#), [ReadAndPrintAttributes.cxx](#), [gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

Referenced by [GetAsDataElement\(\)](#), and [Print\(\)](#).

12.17.4.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
```

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
    unsigned int idx = 0) [inline]
```

Examples

[DeriveSeries.cxx](#), [FixOrientation.cxx](#), [GetSequenceUltrasound.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcm__assert](#), [GetNumberOfValues\(\)](#), and [Internal](#).

Referenced by [operator\[\]\(\)](#), and [operator\[\]\(\)](#).

12.17.4.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
```

```
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

References [gdcm__assert](#), [GetNumberOfValues\(\)](#), and [Internal](#).

12.17.4.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
```

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]
```

Examples

[FixOrientation.cxx](#), [LargeVRDSExplicit.cxx](#), [gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

References [Internal](#).

Referenced by [operator!=\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=\(\)](#), [operator<\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<\(\)](#), [operator==\(\)](#), and [gdcm::Attribute< Group, Element,](#)

12.17.4.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetVM () [inline], [static]
```

12.17.4.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

Examples

[LargeVRDSExplicit.cxx](#).

Referenced by [GetAsDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.17.4.14 operator"!=()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
    const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References [GetNumberOfValues\(\)](#), [GetValues\(\)](#), and [Internal](#).

12.17.4.15 operator<()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References [GetNumberOfValues\(\)](#), [GetValues\(\)](#), and [Internal](#).

12.17.4.16 operator==()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
    const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References [GetNumberOfValues\(\)](#), [GetValues\(\)](#), and [Internal](#).

12.17.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

References [GetValue\(\)](#).

12.17.4.18 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

References [GetValue\(\)](#).

12.17.4.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

References [GetNumberOfValues\(\)](#), [GetTag\(\)](#), and [Internal](#).

12.17.4.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

Examples

[LargeVRDSExplicit.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

References [gdcm::DataSet::GetDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.17.4.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

References [gdcm_assert](#), [gdcm::ByteValue::GetLength\(\)](#), [GetNumberOfValues\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), and [Internal](#).

Referenced by [SetFromDataElement\(\)](#).

12.17.4.22 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

References [gdcmm::assert](#), [gdcmm::ByteValue::GetLength\(\)](#), [GetNumberOfValues\(\)](#), [gdcmm::ByteValue::GetPointer\(\)](#), and [Internal](#).

Referenced by [SetFromDataElement\(\)](#).

12.17.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de) [inline]
```

Examples

[GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [gdcmmrtionplan.cxx](#), and [gdcmmrtplan.cxx](#).

References [gdcmm::assert](#), [gdcmm::DataElement::GetByteValue\(\)](#), [gdcmm::DataElement::GetTag\(\)](#), [GetVR\(\)](#), [gdcmm::DataElement::GetVR\(\)](#), [gdcmm::VR::INVALID](#), [gdcmm::DataElement::IsEmpty\(\)](#), [SetByteValue\(\)](#), [SetByteValueNoSwap\(\)](#), and [gdcmm::VR::UN](#).

Referenced by [Set\(\)](#), and [SetFromDataSet\(\)](#).

12.17.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds) [inline]
```

Examples

[DeriveSeries.cxx](#), [FixOrientation.cxx](#), [ReadAndPrintAttributes.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcmm::DataSet::FindDataElement\(\)](#), [gdcmm::DataSet::GetDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0) [inline]
```

Examples

[CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [FixOrientation.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), and [PatchFile.cxx](#).

References [gdcm_assert](#), [GetNumberOfValues\(\)](#), and [Internal](#).

12.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType) [inline]
```

Examples

[FixOrientation.cxx](#), and [LargeVRDSExplicit.cxx](#).

References [gdcm_assert](#), [GetNumberOfValues\(\)](#), [Internal](#), and [VMType](#).

12.17.5 Member Data Documentation

12.17.5.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by [GetAsDataElement\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [GetValues\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator==\(\)](#), [Print\(\)](#), [SetByteValue\(\)](#), [SetByteValueNoSwap\(\)](#), [SetValue\(\)](#), and [SetValues\(\)](#).

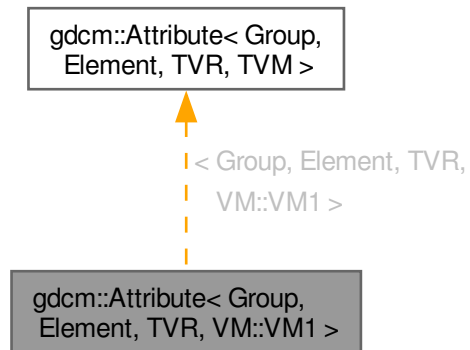
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

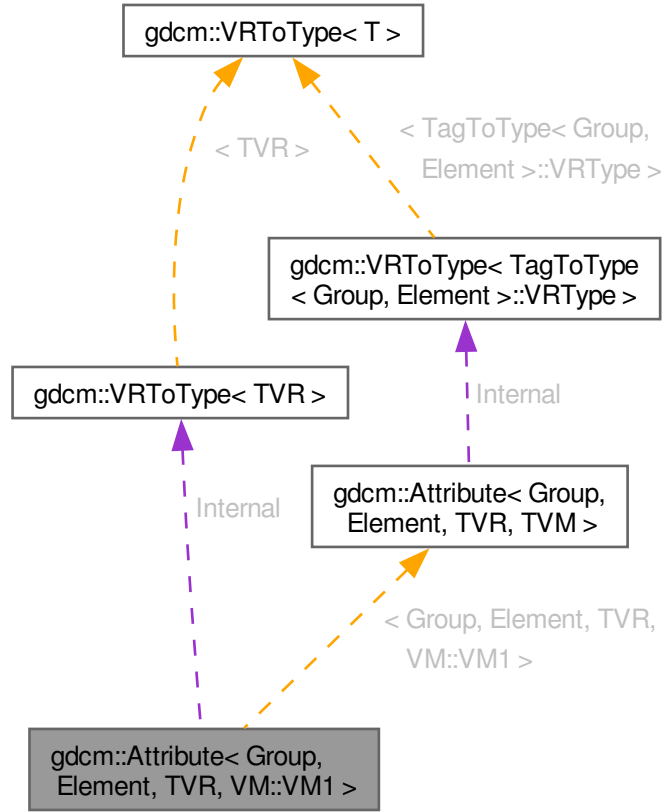
12.18 gdcM::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



Public Types

- enum { `VMType` = `VMToLength<VM::VM1>::Length` }
- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Member Functions

- `GDCM_STATIC_ASSERT` (((((VR::VRTType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))) || TVR &VR::VR_VM1)))
- `GDCM_STATIC_ASSERT` (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))
- `GDCM_STATIC_ASSERT` (((VR::VRTType) TVR &(VR::VRTType)(TagToType< Group, Element >::VRTType)))
- `GDCM_STATIC_ASSERT` (VMToLength< VM::VM1 >::Length==1)
- `DataElement GetAsDataElement () const`

- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) ()
- [ArrayType](#) const & [GetValue](#) () const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [ArrayType](#) [Internal](#)

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.18.1 Member Typedef Documentation

12.18.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType
```


12.18.2 Member Enumeration Documentation

12.18.2.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR>
anonymous enum
```

Enumerator

VMType	
--------	--

12.18.2.2 anonymous enum

anonymous enum

12.18.3 Member Function Documentation

12.18.3.1 GDCM_STATIC_ASSERT() [1/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR
&VR::VR_VM1)) )
```

References [gdcmm::VM::VM1](#), and [gdcmm::VR::VR_VM1](#).

12.18.3.2 GDCM_STATIC_ASSERT() [2/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

References [gdcmm::VM::VM1](#).

12.18.3.3 GDCM_STATIC_ASSERT() [3/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.18.3.4 GDCM_STATIC_ASSERT() [4/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    VMToLength< VM::VM1 >::Length == 1)
```

12.18.3.5 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement () const [inline]
```

References [gdcmm::assert](#), [GetNumberOfValues\(\)](#), [GetVR\(\)](#), [gdcmm::DataElement::GetVR\(\)](#), [Internal](#), [gdcmm::DataElement::SetByteValue\(\)](#), [gdcmm::DataElement::SetVR\(\)](#), [gdcmm::VR::SQ](#), [gdcmm::VR::UI](#), and [gdcmm::VR::VRASCII](#).

12.18.3.6 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM () [inline], [static]
```

12.18.3.7 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR () [inline], [static]
```

12.18.3.8 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues () const [inline]
```

Referenced by [GetAsDataElement\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator==\(\)](#), [SetByteValue\(\)](#), and [SetByteValueNoSwap\(\)](#).

12.18.3.9 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
Tag gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag () [inline], [static]
```

Referenced by [Print\(\)](#).

12.18.3.10 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () [inline]
```

References [Internal](#).

12.18.3.11 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () const [inline]
```

References [Internal](#).

12.18.3.12 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues () const [inline]
```

References [Internal](#).

12.18.3.13 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM () [inline], [static]
```

References [gdcmm::VM::VM1](#).

12.18.3.14 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR () [inline], [static]
```

Referenced by [GetAsDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.18.3.15 operator"!==(

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (
    const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References [GetNumberOfValues\(\)](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#), and [Internal](#).

12.18.3.16 operator<(

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (
    const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References [gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues\(\)](#), [GetNumberOfValues\(\)](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#), and [Internal](#).

12.18.3.17 operator==(

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator== (
    const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References [GetNumberOfValues\(\)](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues\(\)](#), and [Internal](#).

12.18.3.18 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.18.3.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print (
    std::ostream & os) const [inline]
```

References [GetTag\(\)](#), and [Internal](#).

12.18.3.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set (
    DataSet const & ds) [inline]
```

References [gdcmm::DataSet::GetDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.18.3.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

References [gdcmm_assert](#), [gdcmm::ByteValue::GetLength\(\)](#), [GetNumberOfValues\(\)](#), [gdcmm::ByteValue::GetPointer\(\)](#), and [Internal](#).

Referenced by [SetFromDataElement\(\)](#).

12.18.3.22 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

References [gdcmm_assert](#), [gdcmm::ByteValue::GetLength\(\)](#), [GetNumberOfValues\(\)](#), [gdcmm::ByteValue::GetPointer\(\)](#), and [Internal](#).

Referenced by [SetFromDataElement\(\)](#).

12.18.3.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (
    DataElement const & de) [inline]
```

References [gdcm_assert](#), [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [GetVR\(\)](#), [gdcm::DataElement::GetVR\(\)](#), [gdcm::VR::INVALID](#), [gdcm::DataElement::IsEmpty\(\)](#), [SetByteValue\(\)](#), [SetByteValueNoSwap\(\)](#), and [gdcm::VR::UN](#).

Referenced by [Set\(\)](#), and [SetFromDataSet\(\)](#).

12.18.3.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (
    DataSet const & ds) [inline]
```

References [gdcm::DataSet::FindDataElement\(\)](#), [gdcm::DataSet::GetDataElement\(\)](#), and [SetFromDataElement\(\)](#).

12.18.3.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (
    ArrayType v) [inline]
```

References [Internal](#).

12.18.3.26 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType) [inline]
```

12.18.4 Member Data Documentation

12.18.4.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal
```

Referenced by [GetAsDataElement\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [GetValues\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator==\(\)](#), [Print\(\)](#), [SetByteValue\(\)](#), [SetByteValueNoSwap\(\)](#), and [SetValue\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

12.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`:



Collaboration diagram for `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`:



Public Types

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Types inherited from `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >`

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Member Functions

- `GDCM_STATIC_ASSERT` (((`VR::VRTType`) TVR &(`VR::VRTType`)(`TagToType< Group, Element >::VRTType`)))
- `DataElement GetAsDataElement` () const
- unsigned int `GetNumberOfValues` () const
- `ArrayType` & `GetValue` (unsigned int idx=0)
- const `ArrayType` * `GetValues` () const
- `VM` `GetVM` () const
- bool `operator!=` (const `Attribute` &att) const
- bool `operator<` (const `Attribute` &att) const
- bool `operator==` (const `Attribute` &att) const
- `ArrayType` & `operator[]` (unsigned int idx)
- void `Print` (std::ostream &os) const
- void `Set` (`DataSet` const &ds)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetFromDataSet` (`DataSet` const &ds)
- void `SetValue` (`ArrayType` v, unsigned int idx=0)
- void `SetValues` (const `ArrayType` *array, unsigned int numel=`VMType`)

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||((VR::VRType) TVR &VR::VR_VM1)))
- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM_STATIC_ASSERT](#) ((VM::VM1_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [ArrayType](#) [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from
[gdcm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.19.1 Member Typedef Documentation

12.19.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.19.2 Member Enumeration Documentation

12.19.2.1 anonymous enum

anonymous enum

12.19.3 Member Function Documentation

12.19.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.19.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const [inline]
```

12.19.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM () [inline], [static]
```

12.19.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```


12.19.3.5 GetNumberOfValues()

unsigned int [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetNumberOfValues () const [inline]

12.19.3.6 GetTag()

[Tag](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetTag () [inline], [static]

12.19.3.7 GetValue()

[ArrayType](#) & [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetValue (unsigned int idx = 0) [inline]

12.19.3.8 GetValues()

const [ArrayType](#) * [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetValues () const [inline]

12.19.3.9 GetVM()

template<uint16_t Group, uint16_t Element, long long TVR>
[VM](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, [VM::VM1_3](#) >::GetVM () const [inline]

References [gdcmm::VM::VM1_3](#).

12.19.3.10 GetVR()

[VR](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetVR () [inline], [static]

12.19.3.11 operator"!==()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator!=(const [Attribute](#)< Group, [Element](#), TVR, [VM::VM1_3](#) > & att) const [inline]

12.19.3.12 operator<()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator< (const [Attribute](#)< Group, [Element](#), TVR, [VM::VM1_3](#) > & att) const [inline]

12.19.3.13 operator==()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator==(const [Attribute](#)< Group, [Element](#), TVR, [VM::VM1_3](#) > & att) const [inline]

12.19.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.19.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

12.19.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

12.19.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

12.19.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

12.19.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de) [inline]
```

12.19.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds) [inline]
```

12.19.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0) [inline]
```

12.19.3.22 SetValues()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType) [inline]
```

12.19.4 Member Data Documentation

12.19.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

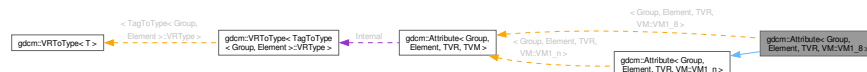
12.20 gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#include <gdcmmAttribute.h>
```

Inheritance diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Collaboration diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Public Types

- enum
- typedef [VRTToType< TVR >::Type](#) ArrayType

Public Types inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- enum
- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

Public Member Functions

- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))([TagToType< Group, Element >::VRType](#))))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- [VM](#) [GetVM](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) (((((([VR::VRType](#)) TVR &[VR::VR_VM1](#)) &&(([VM::VMType](#)) [TagToType< Group, Element >::VMType==VM::VM1](#)))|!(([VR::VRType](#)) TVR &[VR::VR_VM1](#))))
- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))([TagToType< Group, Element >::VRType](#))))
- [GDCM_STATIC_ASSERT](#) (([VM::VM1_n](#) &([VM::VMType](#))([TagToType< Group, Element >::VMType](#))))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from
[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from
[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.20.1 Member Typedef Documentation

12.20.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.20.2 Member Enumeration Documentation

12.20.2.1 anonymous enum

anonymous enum

12.20.3 Member Function Documentation

12.20.3.1 GDCM_STATIC_ASSERT()

`gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GDCM_STATIC_ASSERT (
 ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, `Element` >::VRType)))

12.20.3.2 GetAsDataElement()

`DataElement` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetAsDataElement () const [inline]

12.20.3.3 GetDictVM()

`VM` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetDictVM () [inline], [static]

12.20.3.4 GetDictVR()

`VR` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetDictVR () [inline], [static]

12.20.3.5 GetNumberOfValues()

unsigned int `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetNumberOfValues () const [inline]

12.20.3.6 GetTag()

`Tag` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetTag () [inline], [static]

12.20.3.7 GetValue()

`ArrayType` & `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetValue (
 unsigned int idx = 0) [inline]

12.20.3.8 GetValues()

const `ArrayType` * `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetValues () const [inline]

12.20.3.9 GetVM()

template<uint16_t Group, uint16_t Element, long long TVR>
 `VM` `gdcmm::Attribute`< Group, `Element`, TVR, `VM::VM1_8` >::GetVM () const [inline]

References `gdcmm::VM::VM1_8`.

12.20.3.10 GetVR()

```
VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

12.20.3.11 operator"!=(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(
    const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const [inline]
```

12.20.3.12 operator<(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const [inline]
```

12.20.3.13 operator==(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const [inline]
```

12.20.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.20.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

12.20.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

12.20.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

12.20.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv)    [inline], [protected]
```

12.20.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de)    [inline]
```

12.20.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds)    [inline]
```

12.20.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0)    [inline]
```

12.20.3.22 SetValues()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType)    [inline]
```

12.20.4 Member Data Documentation

12.20.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

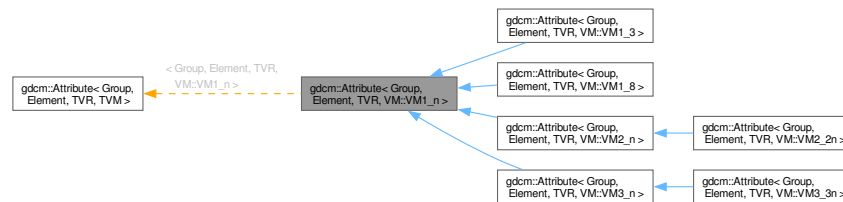
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

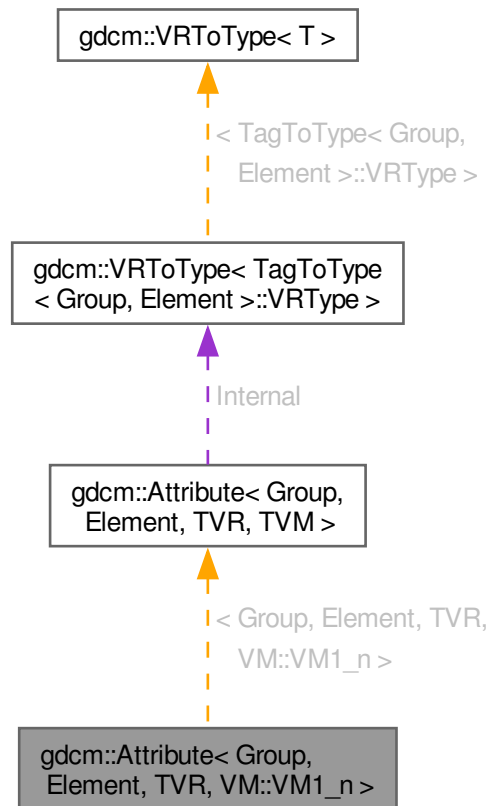
12.21 gdcM::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1_n >:



Collaboration diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1_n >:



Public Types

- enum
- typedef [VRToType](#)< TVR >::Type [ArrayType](#)

Public Member Functions

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM_STATIC_ASSERT](#) ((VM::VM1_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.21.1 Member Typedef Documentation

12.21.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::ArrayType
```

12.21.2 Member Enumeration Documentation

12.21.2.1 anonymous enum

anonymous enum

12.21.3 Constructor & Destructor Documentation

12.21.3.1 Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::Attribute () [inline], [explicit]
```

12.21.3.2 ~Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::~~Attribute () [inline]
```

12.21.4 Member Function Documentation

12.21.4.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::GDCM_STATIC_ASSERT (
    (((VR::VRType) TVR &VR::VR\_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VR
TVR &VR::VR\_VM1)) )
```

References [gdcm::VM::VM1](#), and [gdcm::VR::VR_VM1](#).

12.21.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.21.4.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    (VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

References [gdcm::VM::VM1_n](#).

12.21.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement () const [inline]
```

References [gdcm_assert](#), [GetNumberOfValues\(\)](#), [GetTag\(\)](#), [GetVR\(\)](#), [gdcm::DataElement::GetVR\(\)](#), [gdcm::DataElement::SetByteValue\(\)](#), [gdcm::DataElement::SetVR\(\)](#), [gdcm::VR::SQ](#), [gdcm::VR::UI](#), and [gdcm::VR::VRASCII](#).

12.21.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM () [inline], [static]
```

References [GetVM\(\)](#).

12.21.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR () [inline], [static]
```

12.21.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues () const [inline]
```

Referenced by [GetAsDataElement\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [Print\(\)](#), [SetValue\(\)](#), and [SetValues\(\)](#).

12.21.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag () [inline], [static]
```

Referenced by [GetAsDataElement\(\)](#), [Print\(\)](#), [Set\(\)](#), [SetFromDataElement\(\)](#), and [SetFromDataSet\(\)](#).

12.21.4.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0) [inline]
```

References [gdcmm_assert](#), and [GetNumberOfValues\(\)](#).

Referenced by [operator\[\]\(\)](#), and [operator\[\]\(\)](#).

12.21.4.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0) const [inline]
```

References [gdcmm_assert](#), and [GetNumberOfValues\(\)](#).

12.21.4.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues () const [inline]
```

12.21.4.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM () [inline], [static]
```

References [gdcmm::VM::VM1_n](#).

Referenced by [GetDictVM\(\)](#), and [Print\(\)](#).

12.21.4.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR () [inline], [static]
```

Referenced by [GetAsDataElement\(\)](#), [Print\(\)](#), and [SetFromDataElement\(\)](#).

12.21.4.14 operator"!="()

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!= (
    const Attribute< Group, Element, TVR, VM::VM1_n > & att) const [inline]
```

12.21.4.15 operator<()

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, VM::VM1\_n > & att) const    [inline]
```

12.21.4.16 operator==(())

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, VM::VM1\_n > & att) const    [inline]
```

12.21.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::operator[] (
    unsigned int idx)    [inline]
```

References [GetValue\(\)](#).

12.21.4.18 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::operator[] (
    unsigned int idx) const    [inline]
```

References [GetValue\(\)](#).

12.21.4.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Print (
    std::ostream & os) const    [inline]
```

References [GetNumberOfValues\(\)](#), [GetTag\(\)](#), [GetVM\(\)](#), and [GetVR\(\)](#).

12.21.4.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Set (
    DataSet const & ds)    [inline]
```

References [gdcmm::DataSet::GetDataElement\(\)](#), [GetTag\(\)](#), and [SetFromDataElement\(\)](#).

12.21.4.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

References [gdcm_assert](#), [gdcm::ByteValue::GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), and [SetValues\(\)](#).

Referenced by [SetFromDataElement\(\)](#).

12.21.4.22 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

12.21.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (
    DataElement const & de) [inline]
```

References [gdcm_assert](#), [gdcm::DataElement::GetByteValue\(\)](#), [GetTag\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [GetVR\(\)](#), [gdcm::DataElement::GetVR\(\)](#), [gdcm::DataElement::IsEmpty\(\)](#), and [SetByteValue\(\)](#).

Referenced by [Set\(\)](#), and [SetFromDataSet\(\)](#).

12.21.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (
    DataSet const & ds) [inline]
```

References [gdcm::DataSet::FindDataElement\(\)](#), [gdcm::DataSet::GetDataElement\(\)](#), [GetTag\(\)](#), and [SetFromDataElement\(\)](#).

12.21.4.25 SetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (
    unsigned int numel) [inline]
```

References [SetValues\(\)](#).

12.21.4.26 SetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    ArrayType v) [inline]
```

References [SetValue\(\)](#).

Referenced by [SetValue\(\)](#).

12.21.4.27 SetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    unsigned int idx,
    ArrayType v) [inline]
```

References [gdcm_assert](#), and [GetNumberOfValues\(\)](#).

12.21.4.28 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
    const ArrayType * array,
    unsigned int numel,
    bool own = false) [inline]
```

References [gdcm_assert](#), and [GetNumberOfValues\(\)](#).

Referenced by [SetByteValue\(\)](#), and [SetNumberOfValues\(\)](#).

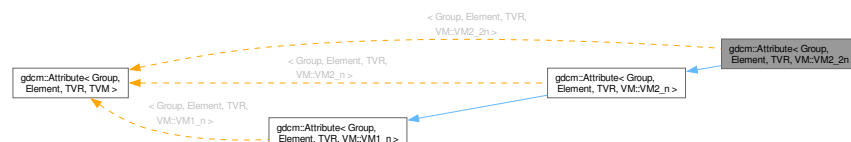
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

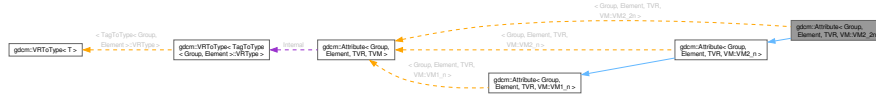
12.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Collaboration diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Public Types

- enum
- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

Public Types inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >](#)

- enum
- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

Public Types inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- enum
- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

Public Member Functions

- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcm::Attribute< Group, Element, TVR, VM::VM2_n >](#)

- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- [VM](#) [GetVM](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) ((((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1)))!((VR::VRType) TVR &VR::VR_VM1))))
- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- [GDCM_STATIC_ASSERT](#) ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Public Attributes inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >](#)

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from
[gdcm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.22.1 Member Typedef Documentation

12.22.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.22.2 Member Enumeration Documentation

12.22.2.1 anonymous enum

anonymous enum

12.22.3 Member Function Documentation

12.22.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM\_STATIC\_ASSERT \(  

    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.22.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement \(\) const [inline]
```

12.22.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM \(\) [inline], [static]
```

12.22.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR \(\) [inline], [static]
```

12.22.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues \(\) const [inline]
```

12.22.3.6 GetTag()

Tag gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline], [static]

12.22.3.7 GetValue()

ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue (
 unsigned int idx = 0) [inline]

12.22.3.8 GetValues()

const ArrayType * gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]

12.22.3.9 GetVM()

template<uint16_t Group, uint16_t Element, long long TVR>
 VM gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM () [inline], [static]

References gdcmm::VM::VM2_2n.

12.22.3.10 GetVR()

VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]

12.22.3.11 operator"!==(

bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(
 const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const [inline]

12.22.3.12 operator<(

bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
 const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const [inline]

12.22.3.13 operator==(

bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
 const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const [inline]

12.22.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.22.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

12.22.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

12.22.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

12.22.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

12.22.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de) [inline]
```

12.22.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds) [inline]
```

12.22.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0) [inline]
```

12.22.3.22 SetValues()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType) [inline]
```

12.22.4 Member Data Documentation

12.22.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

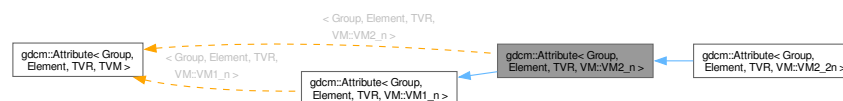
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

12.23 gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmmAttribute.h>
```

Inheritance diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >:



Collaboration diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >:



Public Types

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Types inherited from [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- enum
- typedef [VRToType< TVR >::Type](#) [ArrayType](#)

Public Member Functions

- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))([TagToType< Group, Element >::VRType](#))))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- [VM](#) [GetVM](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) (((((([VR::VRType](#)) TVR &[VR::VR_VM1](#)) &&(([VM::VMType](#)) [TagToType< Group, Element >::VMType==VM::VM1](#)))|!(([VR::VRType](#)) TVR &[VR::VR_VM1](#))))
- [GDCM_STATIC_ASSERT](#) ((([VR::VRType](#)) TVR &([VR::VRType](#))([TagToType< Group, Element >::VRType](#))))
- [GDCM_STATIC_ASSERT](#) (([VM::VM1_n](#) &([VM::VMType](#))([TagToType< Group, Element >::VMType](#))))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from
[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM GetDictVM](#) ()
- static [VR GetDictVR](#) ()
- static [Tag GetTag](#) ()
- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from
[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.23.1 Member Typedef Documentation

12.23.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.23.2 Member Enumeration Documentation

12.23.2.1 anonymous enum

anonymous enum

12.23.3 Member Function Documentation

12.23.3.1 GDCM_STATIC_ASSERT()

`gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GDCM_STATIC_ASSERT (
 ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, `Element` >::VRType)))

12.23.3.2 GetAsDataElement()

`DataElement` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetAsDataElement () const [inline]

12.23.3.3 GetDictVM()

`VM` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetDictVM () [inline], [static]

12.23.3.4 GetDictVR()

`VR` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetDictVR () [inline], [static]

12.23.3.5 GetNumberOfValues()

unsigned int `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetNumberOfValues () const [inline]

12.23.3.6 GetTag()

`Tag` `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetTag () [inline], [static]

12.23.3.7 GetValue()

`ArrayType` & `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetValue (
 unsigned int idx = 0) [inline]

12.23.3.8 GetValues()

const `ArrayType` * `gdcmm::Attribute`< Group, `Element`, TVR, TVM >::GetValues () const [inline]

12.23.3.9 GetVM()

template<uint16_t Group, uint16_t Element, long long TVR>
 `VM` `gdcmm::Attribute`< Group, `Element`, TVR, `VM::VM2_n` >::GetVM () const [inline]

References `gdcmm::VM::VM2_n`.

12.23.3.10 GetVR()

```
VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

12.23.3.11 operator"!=(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(
    const Attribute< Group, Element, TVR, VM::VM2_n > & att) const [inline]
```

12.23.3.12 operator<(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, VM::VM2_n > & att) const [inline]
```

12.23.3.13 operator==(

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, VM::VM2_n > & att) const [inline]
```

12.23.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.23.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

12.23.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

12.23.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

12.23.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv)    [inline], [protected]
```

12.23.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de)    [inline]
```

12.23.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds)    [inline]
```

12.23.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0)    [inline]
```

12.23.3.22 SetValues()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues (
    const ArrayType * array,
    unsigned int numel = VMType)    [inline]
```

12.23.4 Member Data Documentation

12.23.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

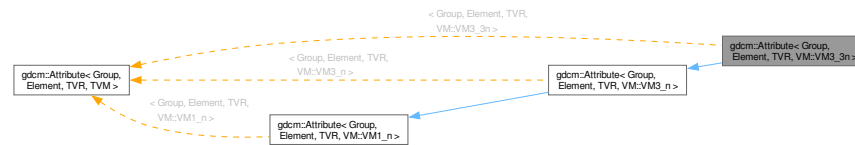
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

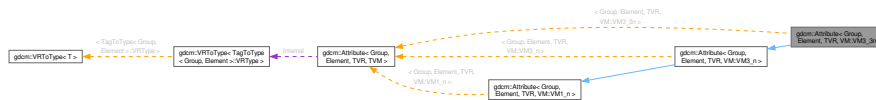
12.24 gdcM::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >:



Collaboration diagram for gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >:



Public Types

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Types inherited from `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >`

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Types inherited from `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >`

- enum
- typedef `VRTToType< TVR >::Type ArrayType`

Public Member Functions

- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >](#)

- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetValue](#) ([ArrayType](#) v, unsigned int idx=0)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel=[VMType](#))

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) ((((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1))))
- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))

- [GDCM_STATIC_ASSERT](#) ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- [DataElement GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from
[gdcm::Attribute< Group, Element, TVR, VM::VM3_n >](#)

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from
[gdcm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Public Attributes inherited from [gdcM::Attribute](#)< Group, Element, TVR, VM::VM3_n >

- [ArrayType Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from [gdcM::Attribute](#)< Group, Element, TVR, VM::VM3_n >

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from [gdcM::Attribute](#)< Group, Element, TVR, VM::VM1_n >

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.24.1 Member Typedef Documentation

12.24.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcM::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.24.2 Member Enumeration Documentation

12.24.2.1 anonymous enum

anonymous enum

12.24.3 Member Function Documentation

12.24.3.1 GDCM_STATIC_ASSERT()

```
gdcM::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR & (VR::VRType)(TagToType< Group, Element >::VRType)) )
```


12.24.3.2 GetAsDataElement()

`DataElement gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const` [inline]

12.24.3.3 GetDictVM()

`VM gdcmm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()` [inline], [static]

12.24.3.4 GetDictVR()

`VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetDictVR ()` [inline], [static]

12.24.3.5 GetNumberOfValues()

`unsigned int gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const` [inline]

12.24.3.6 GetTag()

`Tag gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag ()` [inline], [static]

12.24.3.7 GetValue()

`ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue (`
`unsigned int idx = 0)` [inline]

12.24.3.8 GetValues()

`const ArrayType * gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues () const` [inline]

12.24.3.9 GetVM()

`template<uint16_t Group, uint16_t Element, long long TVR>`
`VM gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ()` [inline], [static]

References `gdcmm::VM::VM3_3n`.

12.24.3.10 GetVR()

`VR gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR ()` [inline], [static]

12.24.3.11 operator"!=()

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(
    const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const    [inline]
```

12.24.3.12 operator<()

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator< (
    const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const    [inline]
```

12.24.3.13 operator==()

```
bool gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(
    const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const    [inline]
```

12.24.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx)    [inline]
```

12.24.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const    [inline]
```

12.24.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds)    [inline]
```

12.24.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv)    [inline], [protected]
```

12.24.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv)    [inline], [protected]
```

12.24.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (  
    DataElement const & de) [inline]
```

12.24.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (  
    DataSet const & ds) [inline]
```

12.24.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (  
    ArrayType v,  
    unsigned int idx = 0) [inline]
```

12.24.3.22 SetValues()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues (  
    const ArrayType * array,  
    unsigned int numel = VMType) [inline]
```

12.24.4 Member Data Documentation

12.24.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

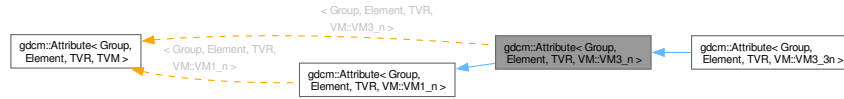
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

12.25 gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

#include <gdcmmAttribute.h>

Inheritance diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >:



Collaboration diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >:



Public Types

- enum
- typedef `VRToType< TVR >::Type` `ArrayType`

Public Types inherited from `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >`

- enum
- typedef `VRToType< TVR >::Type` `ArrayType`

Public Member Functions

- `GDCM_STATIC_ASSERT` (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- `DataElement` `GetAsDataElement` () const
- unsigned int `GetNumberOfValues` () const
- `ArrayType` & `GetValue` (unsigned int idx=0)
- const `ArrayType` * `GetValues` () const
- bool `operator!=` (const `Attribute` &att) const
- bool `operator<` (const `Attribute` &att) const
- bool `operator==` (const `Attribute` &att) const
- `ArrayType` & `operator[]` (unsigned int idx)
- void `Print` (std::ostream &os) const
- void `Set` (`DataSet` const &ds)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetFromDataSet` (`DataSet` const &ds)
- void `SetValue` (`ArrayType` v, unsigned int idx=0)
- void `SetValues` (const `ArrayType` *array, unsigned int numel=`VMType`)

Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- [Attribute](#) ()
- [~Attribute](#) ()
- [GDCM_STATIC_ASSERT](#) (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, [Element](#) >::VMType==VM::VM1))|!((VR::VRType) TVR &VR::VR_VM1))))
- [GDCM_STATIC_ASSERT](#) (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, [Element](#) >::VRType)))
- [GDCM_STATIC_ASSERT](#) ((VM::VM1_n &(VM::VMType)(TagToType< Group, [Element](#) >::VMType)))
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned int [GetNumberOfValues](#) () const
- [ArrayType](#) & [GetValue](#) (unsigned int idx=0)
- [ArrayType](#) const & [GetValue](#) (unsigned int idx=0) const
- const [ArrayType](#) * [GetValues](#) () const
- bool [operator!=](#) (const [Attribute](#) &att) const
- bool [operator<](#) (const [Attribute](#) &att) const
- bool [operator==](#) (const [Attribute](#) &att) const
- [ArrayType](#) & [operator\[\]](#) (unsigned int idx)
- [ArrayType](#) const & [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &os) const
- void [Set](#) ([DataSet](#) const &ds)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetFromDataSet](#) ([DataSet](#) const &ds)
- void [SetNumberOfValues](#) (unsigned int numel)
- void [SetValue](#) ([ArrayType](#) v)
- void [SetValue](#) (unsigned int idx, [ArrayType](#) v)
- void [SetValues](#) (const [ArrayType](#) *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from

[gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- static [VM](#) [GetDictVM](#) ()
- static [VR](#) [GetDictVR](#) ()
- static [Tag](#) [GetTag](#) ()
- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [ArrayType](#) [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

Protected Member Functions inherited from
[gdcm::Attribute< Group, Element, TVR, VM::VM1_n >](#)

- void [SetByteValue](#) (const [ByteValue](#) *bv)
- void [SetByteValueNoSwap](#) (const [ByteValue](#) *bv)

12.25.1 Member Typedef Documentation

12.25.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.25.2 Member Enumeration Documentation

12.25.2.1 anonymous enum

anonymous enum

12.25.3 Member Function Documentation

12.25.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.25.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const [inline]
```

12.25.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM () [inline], [static]
```

12.25.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```

12.25.3.5 GetNumberOfValues()

unsigned int [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetNumberOfValues () const [inline]

12.25.3.6 GetTag()

[Tag](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetTag () [inline], [static]

12.25.3.7 GetValue()

[ArrayType](#) & [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetValue (unsigned int idx = 0) [inline]

12.25.3.8 GetValues()

const [ArrayType](#) * [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetValues () const [inline]

12.25.3.9 GetVM()

template<uint16_t Group, uint16_t Element, long long TVR>
[VM](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, [VM::VM3__n](#) >::GetVM () [inline], [static]

References [gdcmm::VM::VM3__n](#).

12.25.3.10 GetVR()

[VR](#) [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::GetVR () [inline], [static]

12.25.3.11 operator"!==()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator!=(const [Attribute](#)< Group, [Element](#), TVR, [VM::VM3__n](#) > & att) const [inline]

12.25.3.12 operator<()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator< (const [Attribute](#)< Group, [Element](#), TVR, [VM::VM3__n](#) > & att) const [inline]

12.25.3.13 operator==()

bool [gdcmm::Attribute](#)< Group, [Element](#), TVR, TVM >::operator==(const [Attribute](#)< Group, [Element](#), TVR, [VM::VM3__n](#) > & att) const [inline]

12.25.3.14 operator[]()

```
ArrayType & gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx) [inline]
```

12.25.3.15 Print()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os) const [inline]
```

12.25.3.16 Set()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds) [inline]
```

12.25.3.17 SetByteValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv) [inline], [protected]
```

12.25.3.18 SetByteValueNoSwap()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
    const ByteValue * bv) [inline], [protected]
```

12.25.3.19 SetFromDataElement()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de) [inline]
```

12.25.3.20 SetFromDataSet()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds) [inline]
```

12.25.3.21 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    ArrayType v,
    unsigned int idx = 0) [inline]
```


12.25.3.22 SetValue()

```
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue (
    const ArrayType * array,
    unsigned int numel = VMType) [inline]
```

12.25.4 Member Data Documentation

12.25.4.1 Internal

```
ArrayType gdcmm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

The documentation for this class was generated from the following file:

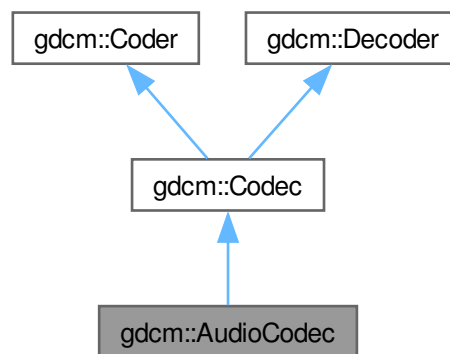
- [gdcmmAttribute.h](#)

12.26 gdcmm::AudioCodec Class Reference

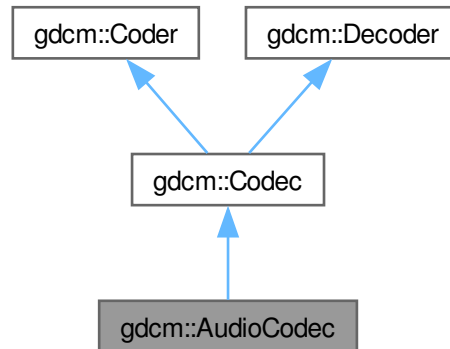
[AudioCodec](#).

```
#include <gdcmmAudioCodec.h>
```

Inheritance diagram for gdcmm::AudioCodec:



Collaboration diagram for `gdcm::AudioCodec`:



Public Member Functions

- [AudioCodec](#) ()
- [~AudioCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override
Return whether this decoder support this transfer syntax (can decode it).
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from [gdcmm::Decoder](#)

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

12.26.1 Detailed Description

[AudioCodec](#).

12.26.2 Constructor & Destructor Documentation

12.26.2.1 AudioCodec()

```
gdcmm::AudioCodec::AudioCodec ()
```

12.26.2.2 ~AudioCodec()

```
gdcmm::AudioCodec::~~AudioCodec () [override]
```

12.26.3 Member Function Documentation

12.26.3.1 CanCode()

```
bool gdcmm::AudioCodec::CanCode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implements [gdcmm::Coder](#).

12.26.3.2 CanDecode()

```
bool gdcmm::AudioCodec::CanDecode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implements [gdcmm::Decoder](#).

12.26.3.3 Decode()

```
bool gdcm::AudioCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

The documentation for this class was generated from the following file:

- [gdcmAudioCodec.h](#)

12.27 gdcm::Base64 Class Reference

Class for [Base64](#).

```
#include <gdcmBase64.h>
```

Public Member Functions

- [Base64](#) (const [Base64](#) &)=delete
- void [operator=](#) (const [Base64](#) &)=delete

Static Public Member Functions

- static size_t [Decode](#) (char *dst, size_t dlen, const char *src, size_t slen)
Decode a base64-formatted buffer.
- static size_t [Encode](#) (char *dst, size_t dlen, const char *src, size_t slen)
Encode a buffer into base64 format.
- static size_t [GetDecodeLength](#) (const char *src, size_t len)
- static size_t [GetEncodeLength](#) (const char *src, size_t srclen)

12.27.1 Detailed Description

Class for [Base64](#).

12.27.2 Constructor & Destructor Documentation

12.27.2.1 Base64()

```
gdcm::Base64::Base64 (
    const Base64 & ) [delete]
```

References [Base64\(\)](#).

Referenced by [Base64\(\)](#), and [operator=\(\)](#).

12.27.3 Member Function Documentation

12.27.3.1 Decode()

```
size_t gdcmm::Base64::Decode (  
    char * dst,  
    size_t dlen,  
    const char * src,  
    size_t slen) [static]
```

Decode a base64-formatted buffer.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be decoded

Returns

0 if not successful, size of decoded otherwise

Examples

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).

12.27.3.2 Encode()

```
size_t gdcmm::Base64::Encode (  
    char * dst,  
    size_t dlen,  
    const char * src,  
    size_t slen) [static]
```

Encode a buffer into base64 format.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be encoded

Returns

0 if not successful, size of encoded otherwise

12.27.3.3 GetDecodeLength()

```
size_t gdcmm::Base64::GetDecodeLength (  
    const char * src,  
    size_t len)    [static]
```

Call this function to obtain the required buffer size

Examples

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).

12.27.3.4 GetEncodeLength()

```
size_t gdcmm::Base64::GetEncodeLength (  
    const char * src,  
    size_t srclen)    [static]
```

Call this function to obtain the required buffer size

12.27.3.5 operator=()

```
void gdcmm::Base64::operator= (  
    const Base64 & )    [delete]
```

References [Base64\(\)](#).

The documentation for this class was generated from the following file:

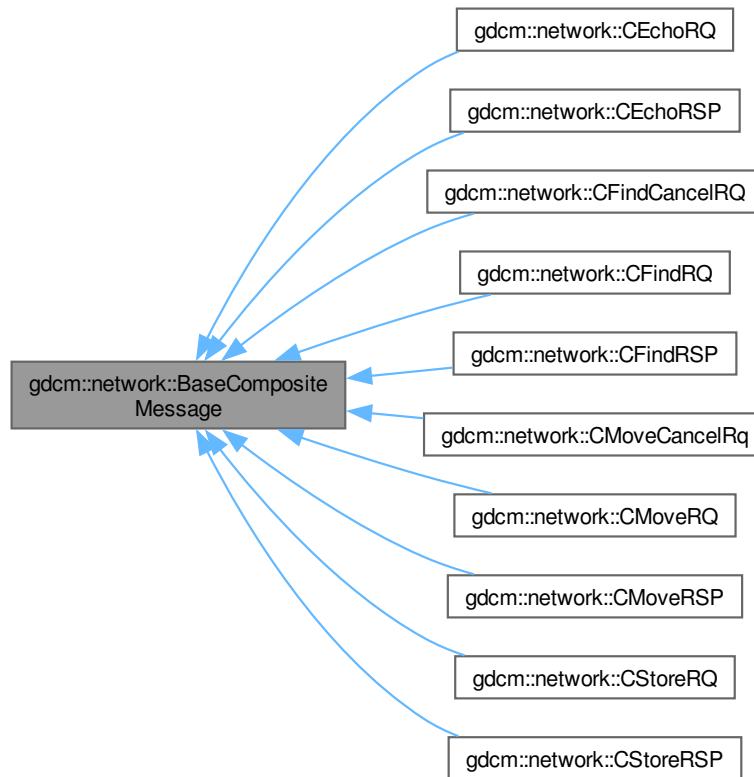
- [gdcmmBase64.h](#)

12.28 gdcmm::network::BaseCompositeMessage Class Reference

[BaseCompositeMessage](#).

```
#include <gdcmmBaseCompositeMessage.h>
```

Inheritance diagram for gdcn::network::BaseCompositeMessage:



Public Member Functions

- virtual `~BaseCompositeMessage()`=default
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

12.28.1 Detailed Description

BaseCompositeMessage.

The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO

- C-FIND
- C-MOVE
- C-GET
- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, `gdcMCompositePDUFactory`.

This is an abstract class. It cannot be instantiated on its own.

12.28.2 Constructor & Destructor Documentation

12.28.2.1 ~BaseCompositeMessage()

```
virtual gdcM::network::BaseCompositeMessage::~~BaseCompositeMessage () [virtual], [default]
```

12.28.3 Member Function Documentation

12.28.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcM::network::BaseCompositeMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery) [pure virtual]
```

Implemented in [gdcM::network::CEchoRQ](#), [gdcM::network::CFindRQ](#), and [gdcM::network::CMoveRQ](#).

The documentation for this class was generated from the following file:

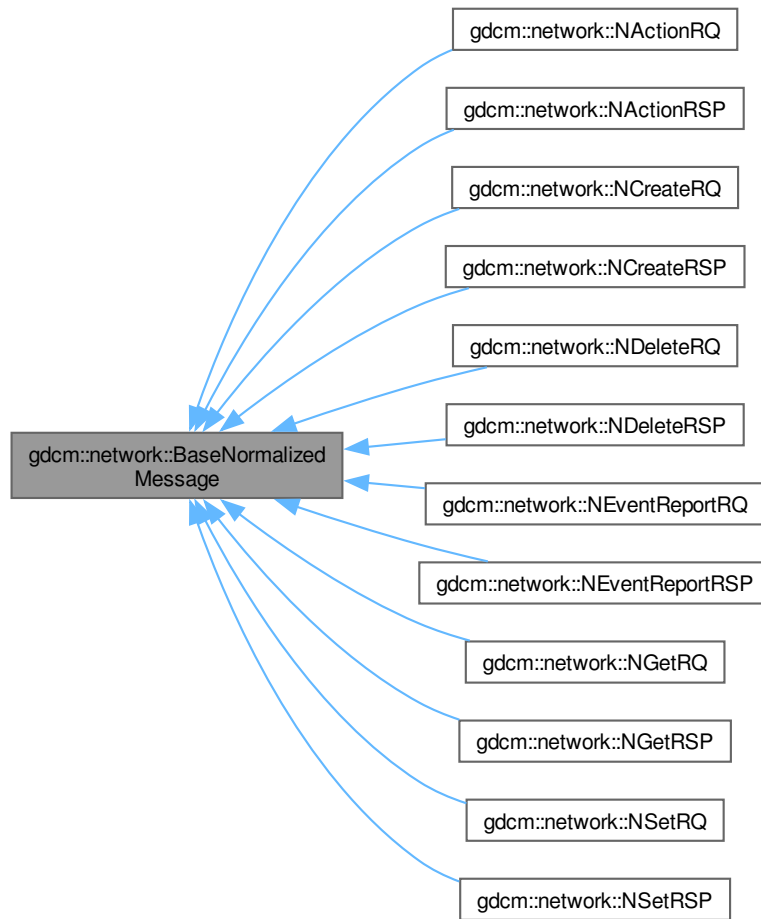
- [gdcMBaseCompositeMessage.h](#)

12.29 gdcM::network::BaseNormalizedMessage Class Reference

[BaseNormalizedMessage](#).

```
#include <gdcMBaseNormalizedMessage.h>
```


Inheritance diagram for gdcmm::network::BaseNormalizedMessage:



Public Member Functions

- virtual `~BaseNormalizedMessage()`=default
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

12.29.1 Detailed Description

BaseNormalizedMessage.

The Normalized events described in section 3.7-2011 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2011 of the standard, and then fill in appropriate values in their datasets.

So, for the five normalized:

- N-ACTION
- N-CREATE
- N-DELETE
- N-EVENT
- N-GET
- N-SET there are a series of messages. However, all of these messages are obtained as part of a PData↔PDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, [gdcmNormalizedMessageFactory.h](#).

This is an abstract class. It cannot be instantiated on its own.

12.29.2 Constructor & Destructor Documentation

12.29.2.1 ~BaseNormalizedMessage()

```
virtual gdcm::network::BaseNormalizedMessage::~BaseNormalizedMessage () [virtual], [default]
```

12.29.3 Member Function Documentation

12.29.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseNormalizedMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [pure virtual]
```

Implemented in [gdcm::network::NActionRQ](#), [gdcm::network::NCreateRQ](#), [gdcm::network::NDeleteRQ](#), [gdcm::network::NEventReportRQ](#), [gdcm::network::NGetRQ](#), and [gdcm::network::NSetRQ](#).

The documentation for this class was generated from the following file:

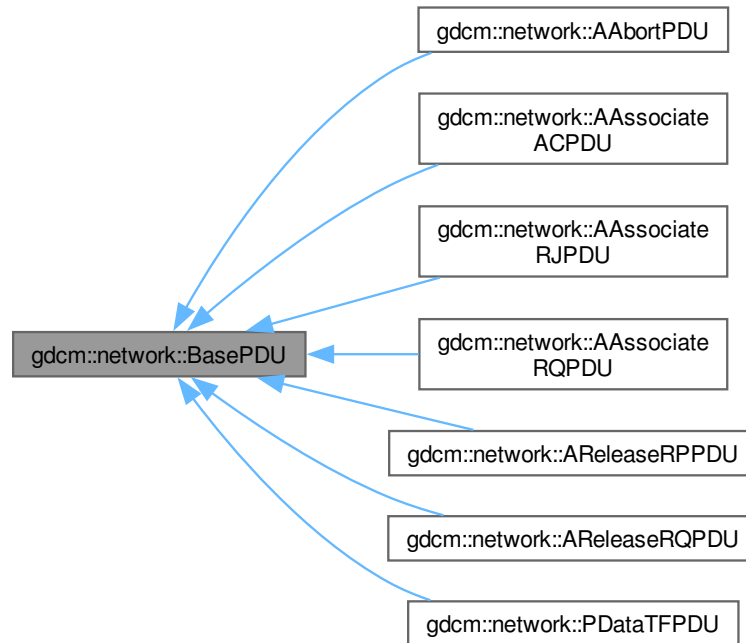
- [gdcmBaseNormalizedMessage.h](#)

12.30 gdcmm::network::BasePDU Class Reference

[BasePDU](#).

```
#include <gdcmmBasePDU.h>
```

Inheritance diagram for gdcmm::network::BasePDU:



Public Member Functions

- virtual [~BasePDU](#) ()=default
- virtual bool [IsLastFragment](#) () const =0
- virtual void [Print](#) (std::ostream &os) const =0
- virtual std::istream & [Read](#) (std::istream &is)=0
- virtual size_t [Size](#) () const =0
- virtual const std::ostream & [Write](#) (std::ostream &os) const =0

12.30.1 Detailed Description

[BasePDU](#).

base class for PDUs

all PDUs start with the first ten bytes as specified: 01 PDU type 02 reserved 3-6 PDU Length (unsigned) 7-10 variable

on some, 7-10 are split (7-8 as protocol version in Associate-RQ, for instance, while associate-rj splits those four bytes differently).

Also common to all the PDUs is their ability to read and write to a stream.

So, let's just get them all bunched together into one (abstract) class, shall we?

Why? 1) so that the [ULEvent](#) can have the PDU stored in it, since the event takes PDUs and not other class structures (other class structures get converted into PDUs) 2) to make reading PDUs in the event loop cleaner

12.30.2 Constructor & Destructor Documentation

12.30.2.1 ~BasePDU()

```
virtual gdcmm::network::BasePDU::~~BasePDU () [virtual], [default]
```

12.30.3 Member Function Documentation

12.30.3.1 IsLastFragment()

```
virtual bool gdcmm::network::BasePDU::IsLastFragment () const [pure virtual]
```

Implemented in [gdcmm::network::AAabortPDU](#), [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::AReleaseRPPDU](#), [gdcmm::network::AReleaseRQPDU](#), and [gdcmm::network::PDataTFPDU](#).

12.30.3.2 Print()

```
virtual void gdcmm::network::BasePDU::Print (
    std::ostream & os) const [pure virtual]
```

Implemented in [gdcmm::network::AAabortPDU](#), [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::AReleaseRPPDU](#), [gdcmm::network::AReleaseRQPDU](#), and [gdcmm::network::PDataTFPDU](#).

12.30.3.3 Read()

```
virtual std::istream & gdcmm::network::BasePDU::Read (
    std::istream & is) [pure virtual]
```

Implemented in [gdcmm::network::AAabortPDU](#), [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::AReleaseRPPDU](#), [gdcmm::network::AReleaseRQPDU](#), and [gdcmm::network::PDataTFPDU](#).

12.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size () const [pure virtual]
```

Implemented in [gdcm::network::AAbortPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::PDataTFPDU](#).

12.30.3.5 Write()

```
virtual const std::ostream & gdcm::network::BasePDU::Write (
    std::ostream & os) const [pure virtual]
```

Implemented in [gdcm::network::AAbortPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::PDataTFPDU](#).

The documentation for this class was generated from the following file:

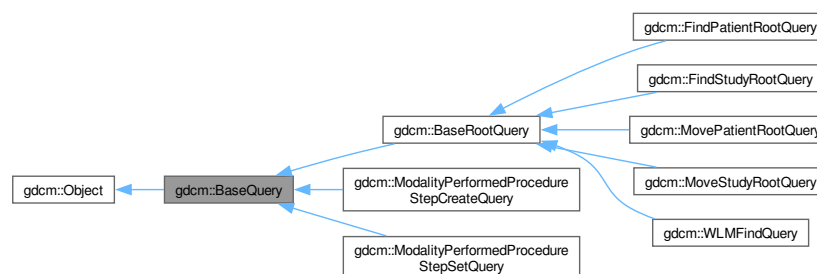
- [gdcmBasePDU.h](#)

12.31 gdcm::BaseQuery Class Reference

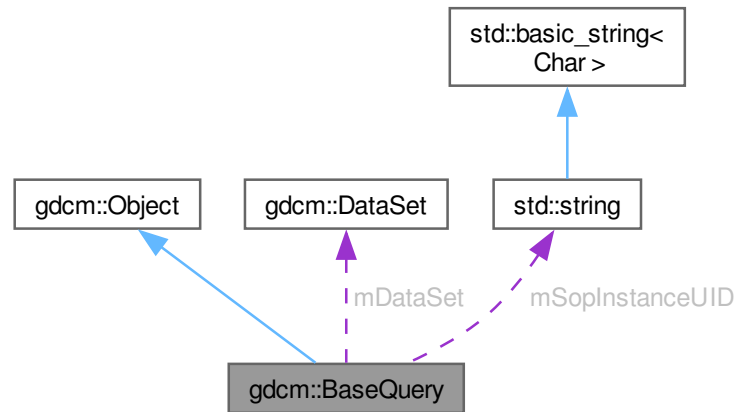
[BaseQuery](#).

```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for gdcm::BaseQuery:



Collaboration diagram for `gdcm::BaseQuery`:



Public Member Functions

- `~BaseQuery ()` override
- `void AddQueryDataSet (const DataSet &ds)`
- `virtual UIDs::TSName GetAbstractSyntaxUID () const =0`
- `DataSet & GetQueryDataSet ()`
- `DataSet const & GetQueryDataSet () const`
Set/Get the internal representation of the query as a `DataSet`.
- `std::string GetSOPInstanceUID () const`
- `void Print (std::ostream &os) const` override
- `void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)`
- `void SetSearchParameter (const Tag &inTag, const std::string &inValue)`
- `void SetSOPInstanceUID (const std::string &iSopInstanceUID)`
- `virtual bool ValidateQuery (bool inStrict=true) const =0`
- `const std::ostream & WriteHelpFile (std::ostream &os)`
- `bool WriteQuery (const std::string &inFileName)`

Public Member Functions inherited from `gdcm::Object`

- `Object ()`
- `Object (const Object &)`
Special requirement for copy/cstor, assignment operator.
- `virtual ~Object ()`
- `void operator= (const Object &)`

Protected Member Functions

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

Friends

- class [QueryFactory](#)

12.31.1 Detailed Description

[BaseQuery](#).

contains: a baseclass which will produce a dataset for all dimse messages

12.31.2 Constructor & Destructor Documentation

12.31.2.1 BaseQuery()

gdcm::BaseQuery::BaseQuery () [protected]

12.31.2.2 ~BaseQuery()

gdcm::BaseQuery::~~BaseQuery () [override]

12.31.3 Member Function Documentation

12.31.3.1 AddQueryDataSet()

void gdcm::BaseQuery::AddQueryDataSet (
 const [DataSet](#) & ds)

12.31.3.2 GetAbstractSyntaxUID()

virtual [UIDs::TSName](#) gdcm::BaseQuery::GetAbstractSyntaxUID () const [pure virtual]

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::ModalityPerformedProcedureStepCreateC](#), [gdcm::ModalityPerformedProcedureStepSetQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

12.31.3.3 GetQueryDataSet() [1/2]

[DataSet](#) & gdcm::BaseQuery::GetQueryDataSet ()

12.31.3.4 GetQueryDataSet() [2/2]

[DataSet](#) const & gdcm::BaseQuery::GetQueryDataSet () const

Set/Get the internal representation of the query as a [DataSet](#).

12.31.3.5 GetSOPInstanceUID()

std::string gdcm::BaseQuery::GetSOPInstanceUID () const [inline]

References [mSopInstanceUID](#).

12.31.3.6 Print()

void gdcm::BaseQuery::Print (
std::ostream & os) const [override], [virtual]

Reimplemented from [gdcm::Object](#).

12.31.3.7 SetSearchParameter() [1/3]

void gdcm::BaseQuery::SetSearchParameter (
const std::string & inKeyword,
const std::string & inValue)

12.31.3.8 SetSearchParameter() [2/3]

void gdcm::BaseQuery::SetSearchParameter (
const [Tag](#) & inTag,
const [DictEntry](#) & inDictEntry,
const std::string & inValue) [protected]

12.31.3.9 SetSearchParameter() [3/3]

```
void gdcmm::BaseQuery::SetSearchParameter (  
    const Tag & inTag,  
    const std::string & inValue)
```

12.31.3.10 SetSOPInstanceUID()

```
void gdcmm::BaseQuery::SetSOPInstanceUID (  
    const std::string & iSopInstanceUID) [inline]
```

References [mSopInstanceUID](#).

12.31.3.11 ValidateQuery()

```
virtual bool gdcmm::BaseQuery::ValidateQuery (  
    bool inStrict = true) const [pure virtual]
```

Implemented in [gdcmm::BaseRootQuery](#), [gdcmm::FindPatientRootQuery](#), [gdcmm::FindStudyRootQuery](#), [gdcmm::ModalityPerformedProcedureStepCreateQuery](#), [gdcmm::ModalityPerformedProcedureStepSetQuery](#), [gdcmm::MovePatientRootQuery](#), [gdcmm::MoveStudyRootQuery](#), and [gdcmm::WLMFindQuery](#).

12.31.3.12 ValidDataSet()

```
bool gdcmm::BaseQuery::ValidDataSet (  
    const DataSet & dataSetToValid,  
    const DataSet & dataSetReference) const [protected]
```

12.31.3.13 WriteHelpFile()

```
const std::ostream & gdcmm::BaseQuery::WriteHelpFile (  
    std::ostream & os)
```

12.31.3.14 WriteQuery()

```
bool gdcmm::BaseQuery::WriteQuery (  
    const std::string & inFileName)
```

12.31.4 Friends And Related Symbol Documentation

12.31.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

12.31.5 Member Data Documentation

12.31.5.1 mDataSet

[DataSet](#) gdcM::BaseQuery::mDataSet [protected]

12.31.5.2 mSopInstanceUID

std::string gdcM::BaseQuery::mSopInstanceUID [protected]

Referenced by [GetSOPInstanceUID\(\)](#), and [SetSOPInstanceUID\(\)](#).

The documentation for this class was generated from the following file:

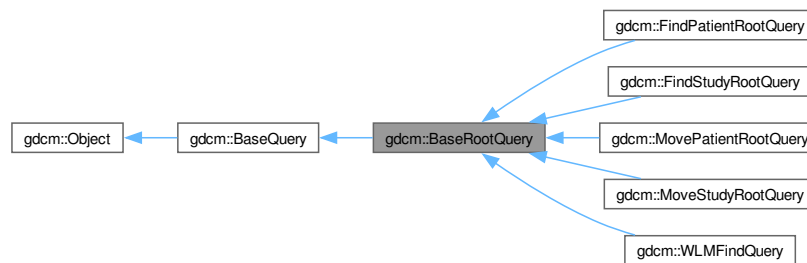
- [gdcMBaseQuery.h](#)

12.32 gdcM::BaseRootQuery Class Reference

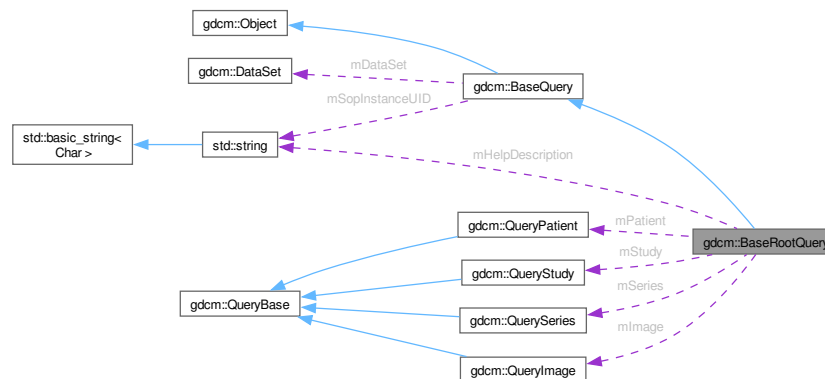
[BaseRootQuery](#).

```
#include <gdcMBaseRootQuery.h>
```

Inheritance diagram for gdcM::BaseRootQuery:



Collaboration diagram for gdcM::BaseRootQuery:



Public Member Functions

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)
- virtual std::vector< [Tag](#) > [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)=0
- virtual void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)=0
- bool [ValidateQuery](#) (bool inStrict=true) const override=0

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- virtual [UIDs::TSName GetAbstractSyntaxUID](#) () const =0
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [QueryBase * Construct](#) (ERootType inRootType, [EQueryLevel](#) qlvel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) ([EQueryLevel](#) ql)

Protected Member Functions

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

Friends

- class [QueryFactory](#)

12.32.1 Detailed Description

[BaseRootQuery](#).

contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root

This class contains the functionality used in patient c-find and c-move queries. [PatientRootQuery](#) and [StudyRootQuery](#) derive from this class.

Namely: 1) list all tags associated with a particular query type 2) produce a query dataset via tag association

Eventually, it can be used to validate a particular dataset type.

The dataset held by this object (or, really, one of its derivatives) should be passed to a c-find or c-move query.

12.32.2 Constructor & Destructor Documentation

12.32.2.1 [BaseRootQuery](#)()

[gdcm::BaseRootQuery::BaseRootQuery](#) () [protected]

12.32.2.2 ~BaseRootQuery()

```
gdcm::BaseRootQuery::~~BaseRootQuery () [override], [default]
```

12.32.3 Member Function Documentation

12.32.3.1 Construct()

```
QueryBase * gdcm::BaseRootQuery::Construct (
    ERootType inRootType,
    EQueryLevel qllevel) [static]
```

12.32.3.2 GetQueryLevelFromQueryRoot()

```
EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (
    ERootType roottype)
```

12.32.3.3 GetQueryLevelFromString()

```
int gdcm::BaseRootQuery::GetQueryLevelFromString (
    const char * str) [static]
```

12.32.3.4 GetQueryLevelString()

```
const char * gdcm::BaseRootQuery::GetQueryLevelString (
    EQueryLevel ql) [static]
```

12.32.3.5 GetTagListByLevel()

```
virtual std::vector< Tag > gdcm::BaseRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel) [pure virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

12.32.3.6 InitializeDataSet()

```
virtual void gdcm::BaseRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel) [pure virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implemented in [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::WLMFindQuery](#).

12.32.3.7 ValidateQuery()

```
bool gdcmm::BaseRootQuery::ValidateQuery (
    bool inStrict = true) const    [override], [pure virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcmm::BaseQuery](#).

Implemented in [gdcmm::FindPatientRootQuery](#), [gdcmm::FindStudyRootQuery](#), [gdcmm::MovePatientRootQuery](#), [gdcmm::MoveStudyRootQuery](#), and [gdcmm::WLMFindQuery](#).

12.32.4 Friends And Related Symbol Documentation

12.32.4.1 QueryFactory

```
friend class QueryFactory    [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

12.32.5 Member Data Documentation

12.32.5.1 mHelpDescription

```
std::string gdcmm::BaseRootQuery::mHelpDescription    [protected]
```

12.32.5.2 mImage

```
QueryImage gdcmm::BaseRootQuery::mImage    [protected]
```

12.32.5.3 mPatient

```
QueryPatient gdcmm::BaseRootQuery::mPatient    [protected]
```

12.32.5.4 mRootType

[ERootType](#) gdcm::BaseRootQuery::mRootType [protected]

12.32.5.5 mSeries

[QuerySeries](#) gdcm::BaseRootQuery::mSeries [protected]

12.32.5.6 mStudy

[QueryStudy](#) gdcm::BaseRootQuery::mStudy [protected]

The documentation for this class was generated from the following file:

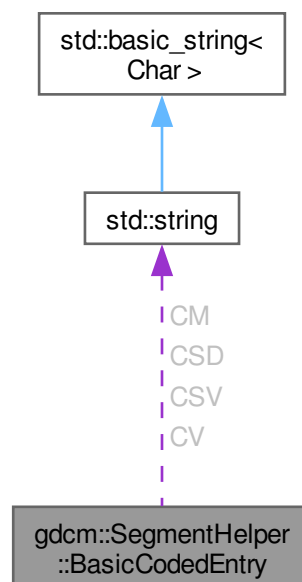
- [gdcmBaseRootQuery.h](#)

12.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference

This structure defines a basic coded entry with all of its attributes.

```
#include <gdcmSegmentHelper.h>
```

Collaboration diagram for gdcm::SegmentHelper::BasicCodedEntry:



Public Member Functions

- [BasicCodedEntry](#) ()
Constructor.
- [BasicCodedEntry](#) (const char *a_CV, const char *a_CSD, const char *a_CM)
constructor which defines type 1 attributes.
- [BasicCodedEntry](#) (const char *a_CV, const char *a_CSD, const char *a_CSV, const char *a_CM)
constructor which defines attributes.
- bool [IsEmpty](#) (const bool checkOptionalAttributes=false) const
Check if each attributes of the basic coded entry is defined.

Public Attributes

- std::string [CM](#)
Coding Scheme [Version](#) attribute.
- std::string [CSD](#)
Code [Value](#) attribute.
- std::string [CSV](#)
Coding Scheme Designator attribute.
- std::string [CV](#)

12.33.1 Detailed Description

This structure defines a basic coded entry with all of its attributes.

See also

PS 3.3 section 8.8.

12.33.2 Constructor & Destructor Documentation

12.33.2.1 BasicCodedEntry() [1/3]

gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry () [inline]

Constructor.

References [CM](#), [CSD](#), [CSV](#), and [CV](#).

12.33.2.2 BasicCodedEntry() [2/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
    const char * a_CV,
    const char * a_CSD,
    const char * a_CM) [inline]
```

constructor which defines type 1 attributes.

References [CM](#), [CSD](#), [CSV](#), and [CV](#).

12.33.2.3 BasicCodedEntry() [3/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
    const char * a_CV,
    const char * a_CSD,
    const char * a_CSV,
    const char * a_CM) [inline]
```

constructor which defines attributes.

References [CM](#), [CSD](#), [CSV](#), and [CV](#).

12.33.3 Member Function Documentation

12.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
    const bool checkOptionalAttributes = false) const
```

Check if each attributes of the basic coded entry is defined.

Parameters

checkOptionalAttributes	Check also type 1C attributes.
-------------------------	--------------------------------

12.33.4 Member Data Documentation

12.33.4.1 CM

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CM
```

Coding Scheme [Version](#) attribute.

Referenced by [BasicCodedEntry\(\)](#), [BasicCodedEntry\(\)](#), and [BasicCodedEntry\(\)](#).

12.33.4.2 CSD

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSD
```

Code [Value](#) attribute.

Referenced by [BasicCodedEntry\(\)](#), [BasicCodedEntry\(\)](#), and [BasicCodedEntry\(\)](#).

12.33.4.3 CSV

`std::string gdcm::SegmentHelper::BasicCodedEntry::CSV`

Coding Scheme Designator attribute.

Referenced by [BasicCodedEntry\(\)](#), [BasicCodedEntry\(\)](#), and [BasicCodedEntry\(\)](#).

12.33.4.4 CV

`std::string gdcm::SegmentHelper::BasicCodedEntry::CV`

Referenced by [BasicCodedEntry\(\)](#), [BasicCodedEntry\(\)](#), and [BasicCodedEntry\(\)](#).

The documentation for this struct was generated from the following file:

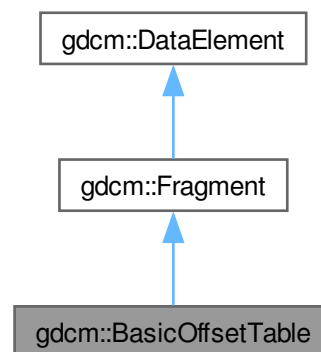
- [gdcmSegmentHelper.h](#)

12.34 gdcm::BasicOffsetTable Class Reference

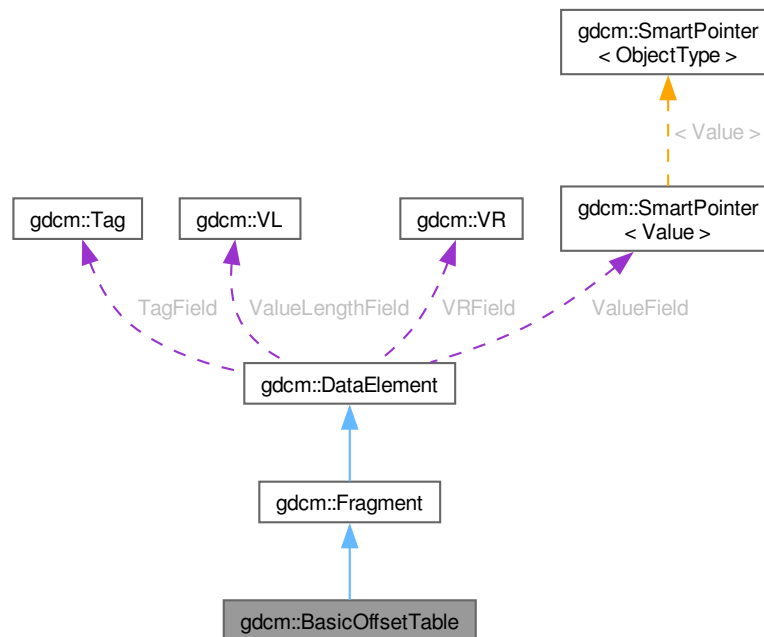
Class to represent a [BasicOffsetTable](#).

```
#include <gdcmBasicOffsetTable.h>
```

Inheritance diagram for `gdcm::BasicOffsetTable`:



Collaboration diagram for gdcm::BasicOffsetTable:



Public Member Functions

- [BasicOffsetTable](#) ()
- [template<typename TSwap> std::istream & Read](#) (std::istream &is)

Public Member Functions inherited from [gdcm::Fragment](#)

- [Fragment](#) ()
- [VL ComputeLength](#) () const
- [VL GetLength](#) () const
- [template<typename TSwap> std::istream & Read](#) (std::istream &is)
- [template<typename TSwap> std::istream & ReadBacktrack](#) (std::istream &is)
- [template<typename TSwap> std::istream & ReadPreValue](#) (std::istream &is)
- [template<typename TSwap> std::istream & ReadValue](#) (std::istream &is)
- [template<typename TSwap> std::ostream & Write](#) (std::ostream &os) const

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
 - Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
 - Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
 [VL](#) [GetLength](#) () const
- [SequenceOffragments](#) * [GetSequenceOffragments](#) ()
- const [SequenceOffragments](#) * [GetSequenceOffragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
 - Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
 - Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOffItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
 - Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
 - Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
 - return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
 std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
 std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
 const std::ostream & [Write](#) (std::ostream &os) const

Friends

- `std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)`

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.34.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

12.34.2 Constructor & Destructor Documentation

12.34.2.1 BasicOffsetTable()

```
gdcm::BasicOffsetTable::BasicOffsetTable () [inline]
```

References [gdcm::Fragment::Fragment\(\)](#).

Referenced by [operator<<](#).

12.34.3 Member Function Documentation

12.34.3.1 Read()

```
template<typename TSwap>
std::istream & gdcm::BasicOffsetTable::Read (
    std::istream & is) [inline]
```

References [gdcm_assert](#), [gdcmAssertAlwaysMacro](#), [gdcm::ParseException::SetLastElement\(\)](#), [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueField](#), and [gdcm::DataElement::ValueLengthField](#).

12.34.4 Friends And Related Symbol Documentation

12.34.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const BasicOffsetTable & val) [friend]
```

References [BasicOffsetTable\(\)](#), [gdcm__assert](#), [gdcm::DataElement::GetByteValue\(\)](#), [operator<<](#), [gdcm::DataElement::ValueField](#) and [gdcm::DataElement::ValueLengthField](#).

Referenced by [operator<<](#).

The documentation for this class was generated from the following file:

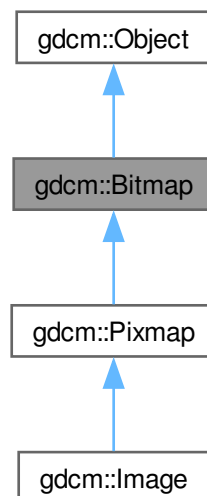
- [gdcmBasicOffsetTable.h](#)

12.35 gdcm::Bitmap Class Reference

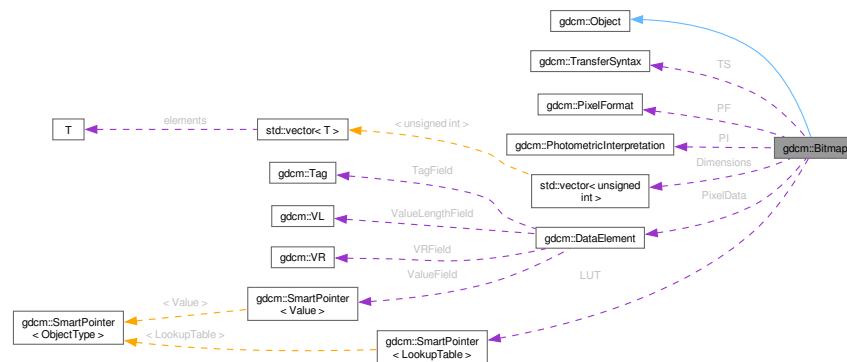
[Bitmap](#) class.

```
#include <gdcmBitmap.h>
```

Inheritance diagram for `gdcm::Bitmap`:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- [Bitmap](#) ()
- [~Bitmap](#) () override
- virtual bool [AreOverlaysInPixelData](#) () const
- void [Clear](#) ()
- bool [GetBuffer](#) (char *buffer) const
Access the raw data.
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int * [GetDimensions](#) () const

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const

INTERNAL do not use.

- unsigned int [GetNumberOfDimensions](#) () const

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const

return the photometric interpretation

- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const

Get/Set [PixelFormat](#).

- unsigned int [GetPlanarConfiguration](#) () const

return the planar configuration

- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const

Return whether or not the image was compressed using a lossy compressor or not.

- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [Print](#) (std::ostream &) const override
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)

Specifically set that the image was compressed using a lossy compression mechanism.

- void [SetLUT](#) ([LookupTable](#) const &lut)

Set/Get LUT.

- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)

Transfer syntax.

- virtual bool [UnusedBitsPresentInPixelData](#) () const

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)

Special requirement for copy/cstor, assignment operator.

- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Protected Types

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions

- bool [ComputeLossyFlag](#) ()
- bool [GetBuffer2](#) (std::ostream &os) const
- bool [TryJPEG2000Codec](#) (char *buffer, bool &lossyflag) const
- bool [TryJPEG2000Codec2](#) (std::ostream &os) const
- bool [TryJPEGCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryJPEGCodec2](#) (std::ostream &os) const
- bool [TryJPEGLSCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryKAKADUCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryPVRGCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryRAWCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryRLECodec](#) (char *buffer, bool &lossyflag) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes

- std::vector< unsigned int > [Dimensions](#)
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- [DataElement](#) [PixelData](#)
- unsigned int [PlanarConfiguration](#)
- [TransferSyntax](#) [TS](#)

Friends

- class [ImageChangeTransferSyntax](#)
- class [PixmapReader](#)

12.35.1 Detailed Description

[Bitmap](#) class.

A bitmap based image. Used as parent for both [IconImage](#) and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

12.35.2 Member Typedef Documentation

12.35.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]
```

12.35.3 Constructor & Destructor Documentation

12.35.3.1 Bitmap()

```
gdcm::Bitmap::Bitmap ()
```

12.35.3.2 ~Bitmap()

gdcm::Bitmap::~~Bitmap () [override]

12.35.4 Member Function Documentation

12.35.4.1 AreOverlaysInPixelData()

virtual bool gdcm::Bitmap::AreOverlaysInPixelData () const [inline], [virtual]

Reimplemented in [gdcm::Pixmap](#).

12.35.4.2 Clear()

void gdcm::Bitmap::Clear ()

12.35.4.3 ComputeLossyFlag()

bool gdcm::Bitmap::ComputeLossyFlag () [protected]

12.35.4.4 GetBuffer()

bool gdcm::Bitmap::GetBuffer (
char * buffer) const

Access the raw data.

Examples

[BasicImageAnonymizer.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#),
[DecompressJPEGFile.cs](#), [GetArray.cs](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

12.35.4.5 GetBuffer2()

bool gdcm::Bitmap::GetBuffer2 (
std::ostream & os) const [protected]

12.35.4.6 GetBufferLength()

unsigned long gdcm::Bitmap::GetBufferLength () const

Return the length of the image after decompression WARNING for palette color: It will NOT take into account the Palette Color thus you need to multiply this length by 3 if computing the size of equivalent RGB image

Examples

[BasicImageAnonymizer.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [GetArray.cs](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

12.35.4.7 GetColumns()

unsigned int gdcm::Bitmap::GetColumns () const [inline]

References [GetDimension\(\)](#).

12.35.4.8 GetDataElement() [1/2]

[DataElement](#) & gdcm::Bitmap::GetDataElement () [inline]

References [PixelData](#).

12.35.4.9 GetDataElement() [2/2]

const [DataElement](#) & gdcm::Bitmap::GetDataElement () const [inline]

Examples

[ExtractIconFromFile.cxx](#).

References [PixelData](#).

12.35.4.10 GetDimension()

unsigned int gdcm::Bitmap::GetDimension (
 unsigned int idx) const

Examples

[BasicImageAnonymizer.cs](#), [DecompressImage.cs](#), and [GetArray.cs](#).

Referenced by [GetColumns\(\)](#), and [GetRows\(\)](#).

12.35.4.11 GetDimensions()

const unsigned int * gdcm::Bitmap::GetDimensions () const

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

Examples

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).

12.35.4.12 GetLUT() [1/2]

[LookupTable](#) & gdcm::Bitmap::GetLUT () [inline]

References [LUT](#).

12.35.4.13 GetLUT() [2/2]

const [LookupTable](#) & gdcm::Bitmap::GetLUT () const [inline]

Examples

[ExtractIconFromFile.cxx](#), [ExtractImageRegionWithLUT.cs](#), and [PrintLUT.cxx](#).

References [LUT](#).

12.35.4.14 GetNeedByteSwap()

bool gdcm::Bitmap::GetNeedByteSwap () const [inline]

INTERNAL do not use.

References [NeedByteSwap](#).

12.35.4.15 GetNumberOfDimensions()

unsigned int gdcm::Bitmap::GetNumberOfDimensions () const

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

Examples

[DecompressImage.cs](#), [GetArray.cs](#), [HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).

12.35.4.16 GetPhotometricInterpretation()

const [PhotometricInterpretation](#) & gdcm::Bitmap::GetPhotometricInterpretation () const

return the photometric interpretation

Examples

[ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

12.35.4.17 GetPixelFormat() [1/2]

[PixelFormat](#) & gdcm::Bitmap::GetPixelFormat () [inline]

References [PF](#).

12.35.4.18 GetPixelFormat() [2/2]

const [PixelFormat](#) & gdcm::Bitmap::GetPixelFormat () const [inline]

Get/Set [PixelFormat](#).

Examples

[ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#),
[GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), and
[threadgdcm.cxx](#).

References [PF](#).

12.35.4.19 GetPlanarConfiguration()

unsigned int gdcm::Bitmap::GetPlanarConfiguration () const

return the planar configuration

12.35.4.20 GetRows()

unsigned int gdcm::Bitmap::GetRows () const [inline]

References [GetDimension\(\)](#).

12.35.4.21 GetTransferSyntax()

const [TransferSyntax](#) & gdcm::Bitmap::GetTransferSyntax () const [inline]

Examples

[ExtractIconFromFile.cxx](#).

References [TS](#).

12.35.4.22 IsEmpty()

bool gdcm::Bitmap::IsEmpty () const [inline]

References [Dimensions](#).

12.35.4.23 IsLossy()

bool gdcm::Bitmap::IsLossy () const

Return whether or not the image was compressed using a lossy compressor or not.

12.35.4.24 IsTransferSyntaxCompatible()

bool gdcm::Bitmap::IsTransferSyntaxCompatible (
 [TransferSyntax](#) const & ts) const

12.35.4.25 Print()

void gdcm::Bitmap::Print (
 std::ostream &) const [override], [virtual]

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Image](#), and [gdcm::Pixmap](#).

Examples

[ExtractIconFromFile.cxx](#).

12.35.4.26 SetColumns()

void gdcm::Bitmap::SetColumns (
 unsigned int col) [inline]

References [SetDimension\(\)](#).

12.35.4.27 SetDataElement()

```
void gdcm::Bitmap::SetDataElement (
    DataElement const & de) [inline]
```

Examples

[BasicImageAnonymizer.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References [PixelData](#).

12.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
    unsigned int idx,
    unsigned int dim)
```

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

Referenced by [SetColumns\(\)](#), and [SetRows\(\)](#).

12.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
    const unsigned int dims[3])
```

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [DecompressImage.cs](#).

12.35.4.30 SetLossyFlag()

```
void gdcm::Bitmap::SetLossyFlag (
    bool f) [inline]
```

Specifically set that the image was compressed using a lossy compression mechanism.

References [LossyFlag](#).

12.35.4.31 SetLUT()

```
void gdcm::Bitmap::SetLUT (  
    LookupTable const & lut) [inline]
```

Set/Get LUT.

References [LUT](#), and [gdcm::Object::SmartPointer](#).

12.35.4.32 SetNeedByteSwap()

```
void gdcm::Bitmap::SetNeedByteSwap (  
    bool b) [inline]
```

References [NeedByteSwap](#).

12.35.4.33 SetNumberOfDimensions()

```
void gdcm::Bitmap::SetNumberOfDimensions (  
    unsigned int dim)
```

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (  
    PhotometricInterpretation const & pi)
```

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (  
    PixelFormat const & pf) [inline]
```

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References [PF](#).

12.35.4.36 SetPlanarConfiguration()

```
void gdcm::Bitmap::SetPlanarConfiguration (  
    unsigned int pc)
```

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

12.35.4.37 SetRows()

```
void gdcm::Bitmap::SetRows (  
    unsigned int rows) [inline]
```

References [SetDimension\(\)](#).

12.35.4.38 SetTransferSyntax()

```
void gdcm::Bitmap::SetTransferSyntax (  
    TransferSyntax const & ts) [inline]
```

Transfer syntax.

Examples

[BasicImageAnonymizer.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImageMultiframe.cs](#),
[DecompressJPEGFile.cs](#), [MergeTwoFiles.cxx](#), and [MpegVideoInfo.cs](#).

References [TS](#).

12.35.4.39 TryJPEG2000Codec()

```
bool gdcm::Bitmap::TryJPEG2000Codec (  
    char * buffer,  
    bool & lossyflag) const [protected]
```

12.35.4.40 TryJPEG2000Codec2()

```
bool gdcm::Bitmap::TryJPEG2000Codec2 (  
    std::ostream & os) const [protected]
```

12.35.4.41 TryJPEGCodec()

```
bool gdcm::Bitmap::TryJPEGCodec (  
    char * buffer,  
    bool & lossyflag) const [protected]
```

12.35.4.42 TryJPEGCodec2()

```
bool gdcm::Bitmap::TryJPEGCodec2 (
    std::ostream & os) const    [protected]
```

12.35.4.43 TryJPEGLSCodec()

```
bool gdcm::Bitmap::TryJPEGLSCodec (
    char * buffer,
    bool & lossyflag) const    [protected]
```

12.35.4.44 TryKAKADUCodec()

```
bool gdcm::Bitmap::TryKAKADUCodec (
    char * buffer,
    bool & lossyflag) const    [protected]
```

12.35.4.45 TryPVRGCodec()

```
bool gdcm::Bitmap::TryPVRGCodec (
    char * buffer,
    bool & lossyflag) const    [protected]
```

12.35.4.46 TryRAWCodec()

```
bool gdcm::Bitmap::TryRAWCodec (
    char * buffer,
    bool & lossyflag) const    [protected]
```

12.35.4.47 TryRLECodec()

```
bool gdcm::Bitmap::TryRLECodec (
    char * buffer,
    bool & lossyflag) const    [protected]
```

12.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcm::Bitmap::UnusedBitsPresentInPixelData () const    [inline], [virtual]
```

Reimplemented in [gdcm::Pixmap](#).

12.35.5 Friends And Related Symbol Documentation

12.35.5.1 ImageChangeTransferSyntax

friend class ImageChangeTransferSyntax [friend]

References [ImageChangeTransferSyntax](#).

Referenced by [ImageChangeTransferSyntax](#).

12.35.5.2 PixmapReader

friend class PixmapReader [friend]

References [PixmapReader](#).

Referenced by [PixmapReader](#).

12.35.6 Member Data Documentation

12.35.6.1 Dimensions

std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

Referenced by [IsEmpty\(\)](#).

12.35.6.2 LossyFlag

bool gdcm::Bitmap::LossyFlag [protected]

Referenced by [SetLossyFlag\(\)](#).

12.35.6.3 LUT

[LUTPtr](#) gdcm::Bitmap::LUT [protected]

Referenced by [GetLUT\(\)](#), [GetLUT\(\)](#), and [SetLUT\(\)](#).

12.35.6.4 NeedByteSwap

bool gdcm::Bitmap::NeedByteSwap [protected]

Referenced by [GetNeedByteSwap\(\)](#), and [SetNeedByteSwap\(\)](#).

12.35.6.5 NumberOfDimensions

unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

12.35.6.6 PF

[PixelFormat](#) gdcm::Bitmap::PF [protected]

Referenced by [GetPixelFormat\(\)](#), [GetPixelFormat\(\)](#), and [SetPixelFormat\(\)](#).

12.35.6.7 PI

[PhotometricInterpretation](#) gdcm::Bitmap::PI [protected]

12.35.6.8 PixelData

[DataElement](#) gdcm::Bitmap::PixelData [protected]

Referenced by [GetDataElement\(\)](#), [GetDataElement\(\)](#), and [SetDataElement\(\)](#).

12.35.6.9 PlanarConfiguration

unsigned int gdcm::Bitmap::PlanarConfiguration [protected]

12.35.6.10 TS

[TransferSyntax](#) gdcm::Bitmap::TS [protected]

Referenced by [GetTransferSyntax\(\)](#), and [SetTransferSyntax\(\)](#).

The documentation for this class was generated from the following file:

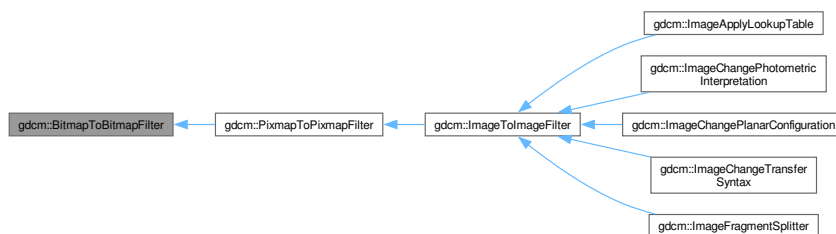
- [gdcmBitmap.h](#)

12.36 gdcm::BitmapToBitmapFilter Class Reference

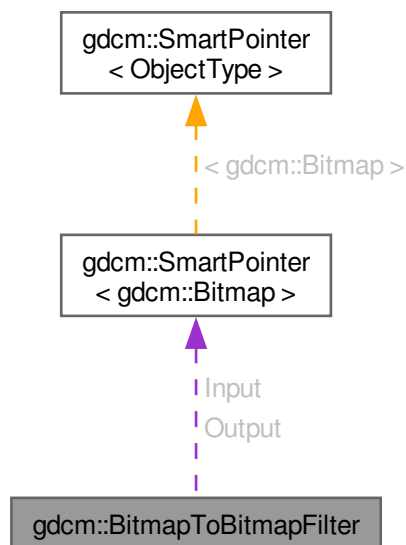
[BitmapToBitmapFilter](#) class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default

- `const Bitmap & GetOutput () const`
Get Output image.
- `const Bitmap & GetOutputAsBitmap () const`
- `void SetInput (const Bitmap &image)`
Set input image.

Protected Attributes

- `SmartPointer< Bitmap > Input`
- `SmartPointer< Bitmap > Output`

12.36.1 Detailed Description

[BitmapToBitmapFilter](#) class.

Super class for all filter taking an image and producing an output image

12.36.2 Constructor & Destructor Documentation

12.36.2.1 [BitmapToBitmapFilter](#)()

`gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ()`

12.36.2.2 `~BitmapToBitmapFilter()`

`gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ()` [default]

12.36.3 Member Function Documentation

12.36.3.1 [GetOutput](#)()

`const Bitmap & gdcm::BitmapToBitmapFilter::GetOutput () const` [inline]

Get Output image.

References [Output](#).

12.36.3.2 [GetOutputAsBitmap](#)()

`const Bitmap & gdcm::BitmapToBitmapFilter::GetOutputAsBitmap () const`

12.36.3.3 SetInput()

```
void gdcm::BitmapToBitmapFilter::SetInput (  
    const Bitmap & image)
```

Set input image.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ExplicitLittleEndian.cs](#), and [StandardizeFiles.cs](#).

12.36.4 Member Data Documentation

12.36.4.1 Input

[SmartPointer<Bitmap>](#) gdcm::BitmapToBitmapFilter::Input [protected]

12.36.4.2 Output

[SmartPointer<Bitmap>](#) gdcm::BitmapToBitmapFilter::Output [protected]

Referenced by [GetOutput\(\)](#).

The documentation for this class was generated from the following file:

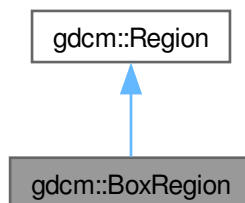
- [gdcmBitmapToBitmapFilter.h](#)

12.37 gdcm::BoxRegion Class Reference

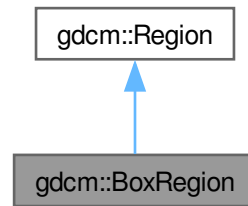
Class for manipulation box region.

```
#include <gdcmBoxRegion.h>
```

Inheritance diagram for gdcm::BoxRegion:



Collaboration diagram for `gdcm::BoxRegion`:



Public Member Functions

- [BoxRegion](#) ()
- [BoxRegion](#) (const [BoxRegion](#) &)
- copy/cstor and al.
- [~BoxRegion](#) () override
- [size_t Area](#) () const override
- compute the area
- [Region * Clone](#) () const override
- [BoxRegion ComputeBoundingBox](#) () override
- Return the Axis-Aligned minimum bounding box for all regions.
- [bool Empty](#) () const override
- return whether this domain is empty:
- [unsigned int GetXMax](#) () const
- [unsigned int GetXMin](#) () const
- Get domain.
- [unsigned int GetYMax](#) () const
- [unsigned int GetYMin](#) () const
- [unsigned int GetZMax](#) () const
- [unsigned int GetZMin](#) () const
- [bool IsValid](#) () const override
- return whether this is valid domain
- [void operator=](#) (const [BoxRegion](#) &)
- [void Print](#) (std::ostream &os=std::cout) const override
- Print.
- [void SetDomain](#) (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)
- Set domain.

Public Member Functions inherited from [gdcm::Region](#)

- [Region](#) ()
- virtual [~Region](#) ()

Static Public Member Functions

- static [BoxRegion BoundingBox](#) ([BoxRegion](#) const &b1, [BoxRegion](#) const &b2)
Helper class to compute the bounding box of two [BoxRegion](#).

12.37.1 Detailed Description

Class for manipulation box region.

This is a very simple implementation of the [Region](#) class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

12.37.2 Constructor & Destructor Documentation

12.37.2.1 [BoxRegion\(\)](#) [1/2]

gdcm::BoxRegion::BoxRegion ()

Referenced by [BoxRegion\(\)](#), [BoundingBox\(\)](#), [ComputeBoundingBox\(\)](#), and [operator=\(\)](#).

12.37.2.2 [~BoxRegion\(\)](#)

gdcm::BoxRegion::~~BoxRegion () [override]

12.37.2.3 [BoxRegion\(\)](#) [2/2]

gdcm::BoxRegion::BoxRegion (
 const [BoxRegion](#) &)

copy/cstor and al.

References [BoxRegion\(\)](#).

12.37.3 Member Function Documentation

12.37.3.1 [Area\(\)](#)

size_t gdcm::BoxRegion::Area () const [override], [virtual]

compute the area

Implements [gdcm::Region](#).

12.37.3.2 BoundingBox()

```
BoxRegion gdcM::BoxRegion::BoundingBox (
    BoxRegion const & b1,
    BoxRegion const & b2) [static]
```

Helper class to compute the bounding box of two [BoxRegion](#).

References [BoxRegion\(\)](#).

12.37.3.3 Clone()

```
Region * gdcM::BoxRegion::Clone () const [override], [virtual]
```

Implements [gdcM::Region](#).

References [gdcM::Region::Region\(\)](#).

12.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcM::BoxRegion::ComputeBoundingBox () [override], [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcM::Region](#).

References [BoxRegion\(\)](#).

12.37.3.5 Empty()

```
bool gdcM::BoxRegion::Empty () const [override], [virtual]
```

return whether this domain is empty:

Implements [gdcM::Region](#).

12.37.3.6 GetXMax()

```
unsigned int gdcM::BoxRegion::GetXMax () const
```

12.37.3.7 GetXMin()

```
unsigned int gdcM::BoxRegion::GetXMin () const
```

Get domain.

12.37.3.8 GetYMax()

unsigned int gdcm::BoxRegion::GetYMax () const

12.37.3.9 GetYMin()

unsigned int gdcm::BoxRegion::GetYMin () const

12.37.3.10 GetZMax()

unsigned int gdcm::BoxRegion::GetZMax () const

12.37.3.11 GetZMin()

unsigned int gdcm::BoxRegion::GetZMin () const

12.37.3.12 IsValid()

bool gdcm::BoxRegion::IsValid () const [override], [virtual]

return whether this is valid domain

Implements [gdcm::Region](#).

12.37.3.13 operator=()

void gdcm::BoxRegion::operator= (
 const [BoxRegion](#) &)

References [BoxRegion\(\)](#).

12.37.3.14 Print()

void gdcm::BoxRegion::Print (
 std::ostream & os = std::cout) const [override], [virtual]

Print.

Reimplemented from [gdcm::Region](#).

12.37.3.15 SetDomain()

```
void gdcM::BoxRegion::SetDomain (  
    unsigned int xmin,  
    unsigned int xmax,  
    unsigned int ymin,  
    unsigned int ymax,  
    unsigned int zmin,  
    unsigned int zmax)
```

Set domain.

Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

The documentation for this class was generated from the following file:

- [gdcMBoxRegion.h](#)

12.38 gdcM::ByteBuffer Class Reference

[ByteBuffer](#).

```
#include <gdcMByteBuffer.h>
```

Public Member Functions

- [ByteBuffer](#) ()
- char * [Get](#) (int len)
- const char * [GetStart](#) () const
- void [ShiftEnd](#) (int len)
- void [UpdatePosition](#) ()

12.38.1 Detailed Description

[ByteBuffer](#).

Detailed description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

12.38.2 Constructor & Destructor Documentation

12.38.2.1 ByteBuffer()

gdcm::ByteBuffer::ByteBuffer () [inline]

12.38.3 Member Function Documentation

12.38.3.1 Get()

char * gdcm::ByteBuffer::Get (
int len) [inline]

References [gdcm_assert](#).

12.38.3.2 GetStart()

const char * gdcm::ByteBuffer::GetStart () const [inline]

12.38.3.3 ShiftEnd()

void gdcm::ByteBuffer::ShiftEnd (
int len) [inline]

12.38.3.4 UpdatePosition()

void gdcm::ByteBuffer::UpdatePosition () [inline]

The documentation for this class was generated from the following file:

- [gdcmByteBuffer.h](#)

12.39 gdcm::ByteSwap< T > Class Template Reference

[ByteSwap](#).

```
#include <gdcmByteSwap.h>
```

Static Public Member Functions

- static void [Swap](#) (T &p)
- static void [SwapFromSwapCodeIntoSystem](#) (T &p, [SwapCode](#) const &sc)
- static void [SwapRange](#) (T *p, unsigned int num)
- static void [SwapRangeFromSwapCodeIntoSystem](#) (T *p, [SwapCode](#) const &sc, std::streamoff num)
- static bool [SystemIsBigEndian](#) ()
- static bool [SystemIsLittleEndian](#) ()

12.39.1 Detailed Description

```
template<class T>
class gdcmm::ByteSwap< T >
```

[ByteSwap](#).

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian).
 TODO: bswap_32 / bswap_64 ...

12.39.2 Member Function Documentation

12.39.2.1 Swap()

```
template<class T>
void gdcmm::ByteSwap< T >::Swap (
    T & p) [static]
```

12.39.2.2 SwapFromSwapCodeIntoSystem()

```
template<class T>
void gdcmm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (
    T & p,
    SwapCode const & sc) [static]
```

Examples

[TestByteSwap.cxx](#).

12.39.2.3 SwapRange()

```
template<class T>
void gdcmm::ByteSwap< T >::SwapRange (
    T * p,
    unsigned int num) [static]
```

12.39.2.4 SwapRangeFromSwapCodeIntoSystem()

```
template<class T>
void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (
    T * p,
    SwapCode const & sc,
    std::streamoff num) [static]
```

Examples

[TestByteSwap.cxx](#).

12.39.2.5 SystemIsBigEndian()

```
template<class T>
bool gdcm::ByteSwap< T >::SystemIsBigEndian () [static]
```

Query the machine Endian-ness.

Examples

[TestByteSwap.cxx](#).

12.39.2.6 SystemIsLittleEndian()

```
template<class T>
bool gdcm::ByteSwap< T >::SystemIsLittleEndian () [static]
```

Examples

[TestByteSwap.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmByteSwap.h](#)

12.40 gdcm::ByteSwapFilter Class Reference

[ByteSwapFilter](#).

```
#include <gdcmByteSwapFilter.h>
```

Public Member Functions

- [ByteSwapFilter](#) (const [ByteSwapFilter](#) &)=delete
- [ByteSwapFilter](#) ([DataSet](#) &ds)
- [~ByteSwapFilter](#) ()=default
- bool [ByteSwap](#) ()
- [ByteSwapFilter](#) & [operator=](#) (const [ByteSwapFilter](#) &)=delete
- void [SetByteSwapTag](#) (bool b)

12.40.1 Detailed Description

[ByteSwapFilter](#).

In place byte-swapping of a dataset FIXME: FL status ??

12.40.2 Constructor & Destructor Documentation

12.40.2.1 [ByteSwapFilter](#)() [1/2]

```
gdcmm::ByteSwapFilter::ByteSwapFilter (
    DataSet & ds) [inline]
```

Referenced by [ByteSwapFilter\(\)](#), and [operator=\(\)](#).

12.40.2.2 [~ByteSwapFilter](#)()

```
gdcmm::ByteSwapFilter::~~ByteSwapFilter () [default]
```

12.40.2.3 [ByteSwapFilter](#)() [2/2]

```
gdcmm::ByteSwapFilter::ByteSwapFilter (
    const ByteSwapFilter & ) [delete]
```

References [ByteSwapFilter\(\)](#).

12.40.3 Member Function Documentation

12.40.3.1 [ByteSwap](#)()

```
bool gdcmm::ByteSwapFilter::ByteSwap ()
```

Referenced by [gdcmm::Item::Read\(\)](#).

12.40.3.2 operator=()

```
ByteSwapFilter & gdcm::ByteSwapFilter::operator= (  
    const ByteSwapFilter & ) [delete]
```

References [ByteSwapFilter\(\)](#).

12.40.3.3 SetByteSwapTag()

```
void gdcm::ByteSwapFilter::SetByteSwapTag (  
    bool b) [inline]
```

Referenced by [gdcm::Item::Read\(\)](#).

The documentation for this class was generated from the following file:

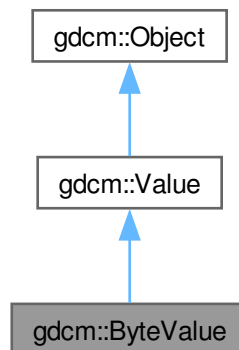
- [gdcmByteSwapFilter.h](#)

12.41 gdcm::ByteValue Class Reference

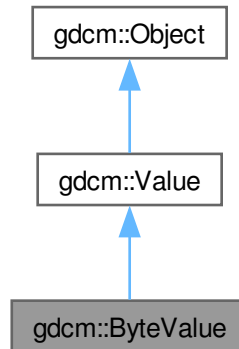
Class to represent binary value (array of bytes).

```
#include <gdcmByteValue.h>
```

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for `gdcm::ByteValue`:



Public Member Functions

- `ByteValue` (const char *array=nullptr, `VL` const &vl=0)
- `ByteValue` (std::vector< char > &v)
- `~ByteValue` () override
- void `Append` (`ByteValue` const &bv)
- void `Clear` () override
- `VL ComputeLength` () const
- void `Fill` (char c)
- bool `GetBuffer` (char *buffer, unsigned long length) const
- `VL GetLength` () const override
- const char * `GetPointer` () const
- void * `GetVoidPointer` ()
- const void * `GetVoidPointer` () const
- bool `IsEmpty` () const
- bool `IsPrintable` (`VL` length) const

Checks whether a '`ByteValue`' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...
- `operator const std::vector< char > &` () const
- `ByteValue & operator=` (const `ByteValue` &val)
- bool `operator==` (const `ByteValue` &val) const
- bool `operator==` (const `Value` &val) const override
- void `PrintASCII` (std::ostream &os, `VL` maxlength) const
- void `PrintASCHXML` (std::ostream &os) const
- void `PrintGroupLength` (std::ostream &os)
- void `PrintHex` (std::ostream &os, `VL` maxlength) const
- void `PrintHexXML` (std::ostream &os) const
- void `PrintPNXML` (std::ostream &os) const
- template<typename TSwap>
std::istream & `Read` (std::istream &is)

- `template<typename TSwap, typename TType>`
`std::istream & Read (std::istream &is, bool readvalues=true)`
- `void SetLength (VL vl) override`
- `template<typename TSwap>`
`std::ostream const & Write (std::ostream &os) const`
- `template<typename TSwap, typename TType>`
`std::ostream const & Write (std::ostream &os) const`
- `bool WriteBuffer (std::ostream &os) const`

Public Member Functions inherited from [gdcm::Value](#)

- [Value](#) ()=default
- [~Value](#) () override=default

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Protected Member Functions

- void [Print](#) (std::ostream &os) const override
- void [SetLengthOnly](#) (VL vl) override

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.41.1 Detailed Description

Class to represent binary value (array of bytes).

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.41.2 Constructor & Destructor Documentation

12.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (  
    const char * array = nullptr,  
    VL const & vl = 0) [inline]
```

Referenced by [Append\(\)](#), [operator=\(\)](#), [operator==\(\)](#), and [operator==\(\)](#).

12.41.2.2 ByteValue() [2/2]

```
gdcm::ByteValue::ByteValue (  
    std::vector< char > & v) [inline]
```

Warning

casting to uint32_t

12.41.2.3 ~ByteValue()

```
gdcm::ByteValue::~ByteValue () [inline], [override]
```

12.41.3 Member Function Documentation

12.41.3.1 Append()

```
void gdcm::ByteValue::Append (  
    ByteValue const & bv)
```

References [ByteValue\(\)](#).

12.41.3.2 Clear()

```
void gdcm::ByteValue::Clear () [inline], [override], [virtual]
```

Implements [gdcm::Value](#).

12.41.3.3 ComputeLength()

```
VL gdcm::ByteValue::ComputeLength () const [inline]
```

Referenced by [gdcm::Fragment::Write\(\)](#).

12.41.3.4 Fill()

```
void gdcm::ByteValue::Fill (
    char c)    [inline]
```

Examples

[DuplicatePCDE.cxx](#).

12.41.3.5 GetBuffer()

```
bool gdcm::ByteValue::GetBuffer (
    char * buffer,
    unsigned long length) const
```

Examples

[ExtractEncapsulatedFile.cs](#), and [FixJAIBugJPEGLS.cxx](#).

12.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength () const    [inline], [override], [virtual]
```

Implements [gdcm::Value](#).

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::CSAElement::operator<<](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< TVR, TVM >::gdcm::Element< TVR, VM::VM1_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetNoSwap\(\)](#), [gdcm::Element< TVR, TVM >::SetNoSwap\(\)](#), [gdcm::Element< TVR, VM::VM1_n >::SetNoSwap\(\)](#), and [gdcm::Fragment::Write\(\)](#).

12.41.3.7 GetPointer()

```
const char * gdcm::ByteValue::GetPointer () const [inline]
```

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::CSAElement::operator<<](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< TVR, TVM >::Set\(\)](#), [gdcm::Element< TVR, VM::VM1_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetNoSwap\(\)](#), and [gdcm::Element< TVR, VM::VM1_n >::SetNoSwap\(\)](#).

12.41.3.8 GetVoidPointer() [1/2]

```
void * gdcm::ByteValue::GetVoidPointer () [inline]
```

12.41.3.9 GetVoidPointer() [2/2]

```
const void * gdcm::ByteValue::GetVoidPointer () const [inline]
```

Examples

[FixBrokenJ2K.cxx](#).

Referenced by [Read\(\)](#), and [gdcm::Element< TVR, VM::VM1_n >::Set\(\)](#).

12.41.3.10 IsEmpty()

```
bool gdcm::ByteValue::IsEmpty () const [inline]
```

References [gdcm_assert](#).

12.41.3.11 IsPrintable()

```
bool gdcm::ByteValue::IsPrintable (
    VL length) const [inline]
```

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...

References [gdcm_assert](#).

Referenced by [Print\(\)](#).

12.41.3.12 operator const std::vector< char > &()

```
gdcmm::ByteValue::operator const std::vector< char > & () const    [inline]
```

12.41.3.13 operator=()

```
ByteValue & gdcmm::ByteValue::operator= (  
    const ByteValue & val)    [inline]
```

References [ByteValue\(\)](#).

12.41.3.14 operator==() [1/2]

```
bool gdcmm::ByteValue::operator== (  
    const ByteValue & val) const    [inline]
```

References [ByteValue\(\)](#).

12.41.3.15 operator==() [2/2]

```
bool gdcmm::ByteValue::operator== (  
    const Value & val) const    [inline], [override], [virtual]
```

Implements [gdcmm::Value](#).

References [ByteValue\(\)](#), and [gdcmm::Value::Value\(\)](#).

12.41.3.16 Print()

```
void gdcmm::ByteValue::Print (  
    std::ostream & os) const    [inline], [override], [protected], [virtual]
```

Reimplemented from [gdcmm::Object](#).

References [IsPrintable\(\)](#).

12.41.3.17 PrintASCII()

```
void gdcmm::ByteValue::PrintASCII (  
    std::ostream & os,  
    VL maxlength) const
```

12.41.3.18 PrintASCIIXML()

```
void gdcM::ByteValue::PrintASCIIXML (
    std::ostream & os) const
```

12.41.3.19 PrintGroupLength()

```
void gdcM::ByteValue::PrintGroupLength (
    std::ostream & os) [inline]
```

References [gdcM_assert](#).

12.41.3.20 PrintHex()

```
void gdcM::ByteValue::PrintHex (
    std::ostream & os,
    VL maxlength) const
```

12.41.3.21 PrintHexXML()

```
void gdcM::ByteValue::PrintHexXML (
    std::ostream & os) const
```

12.41.3.22 PrintPNXML()

```
void gdcM::ByteValue::PrintPNXML (
    std::ostream & os) const
```

To Print Values in Native DICOM format

12.41.3.23 Read() [1/2]

```
template<typename TSwap>
std::istream & gdcM::ByteValue::Read (
    std::istream & is) [inline]
```

References [Read\(\)](#).

12.41.3.24 Read() [2/2]

```
template<typename TSwap, typename TType>
std::istream & gdcM::ByteValue::Read (
    std::istream & is,
    bool readvalues = true) [inline]
```

References [gdcM_assert](#), and [GetVoidPointer\(\)](#).

Referenced by [Read\(\)](#).

12.41.3.25 SetLength()

```
void gdcm::ByteValue::SetLength (  
    VL vl) [override], [virtual]
```

Implements [gdcm::Value](#).

12.41.3.26 SetLengthOnly()

```
void gdcm::ByteValue::SetLengthOnly (  
    VL vl) [inline], [override], [protected], [virtual]
```

Reimplemented from [gdcm::Value](#).

12.41.3.27 Write() [1/2]

```
template<typename TSwap>  
std::ostream const & gdcm::ByteValue::Write (  
    std::ostream & os) const [inline]
```

References [Write\(\)](#).

12.41.3.28 Write() [2/2]

```
template<typename TSwap, typename TType>  
std::ostream const & gdcm::ByteValue::Write (  
    std::ostream & os) const [inline]
```

References [gdcm_assert](#).

Referenced by [Write\(\)](#), and [gdcm::Fragment::Write\(\)](#).

12.41.3.29 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (  
    std::ostream & os) const [inline]
```

References [gdcm_assert](#).

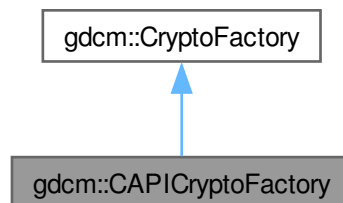
The documentation for this class was generated from the following file:

- [gdcmByteValue.h](#)

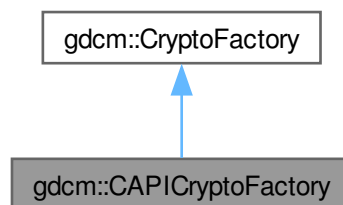
12.42 gdcM::CAPICryptoFactory Class Reference

```
#include <gdcMCAPICryptoFactory.h>
```

Inheritance diagram for gdcM::CAPICryptoFactory:



Collaboration diagram for gdcM::CAPICryptoFactory:



Public Member Functions

- [CAPICryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()

Additional Inherited Members

Public Types inherited from [gdcM::CryptoFactory](#)

- enum [CryptoLib](#) {
 [DEFAULT](#) = 0 ,
 [OPENSSL](#) = 1 ,
 [CAPI](#) = 2 ,
 [OPENSSL7](#) = 3 }

Static Public Member Functions inherited from [gdcm::CryptoFactory](#)

- static [CryptoFactory](#) * [GetFactoryInstance](#) ([CryptoLib](#) id=DEFAULT)

Protected Member Functions inherited from [gdcm::CryptoFactory](#)

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

12.42.1 Constructor & Destructor Documentation

12.42.1.1 CAPICryptoFactory()

[gdcm::CAPICryptoFactory::CAPICryptoFactory](#) (
 [CryptoLib](#) id)

Referenced by [CreateCMSProvider\(\)](#).

12.42.2 Member Function Documentation

12.42.2.1 CreateCMSProvider()

[CryptographicMessageSyntax](#) * [gdcm::CAPICryptoFactory::CreateCMSProvider](#) () [virtual]

Implements [gdcm::CryptoFactory](#).

References [CAPICryptoFactory\(\)](#).

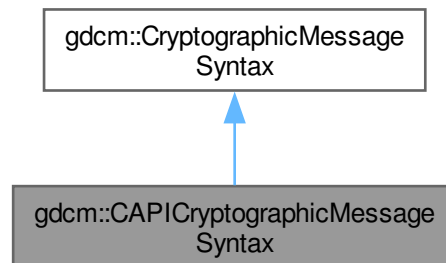
The documentation for this class was generated from the following file:

- [gdcmCAPICryptoFactory.h](#)

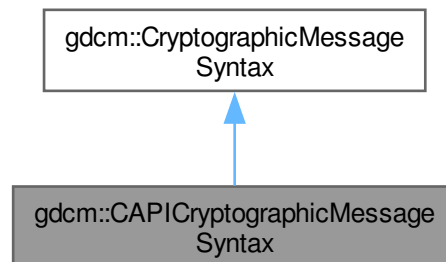
12.43 gdcm::CAPICryptographicMessageSyntax Class Reference

#include <gdcmCAPICryptographicMessageSyntax.h>

Inheritance diagram for gdcm::CAPICryptographicMessageSyntax:



Collaboration diagram for gdcm::CAPICryptographicMessageSyntax:



Public Member Functions

- [CAPICryptographicMessageSyntax](#) ()
- [~CAPICryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
decrypt content from a CMS envelopedData structure
- bool [Encrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
create a CMS envelopedData structure
- [CipherTypes](#) [GetCipherType](#) () const

- bool [GetInitialized](#) () const
- bool [ParseCertificateFile](#) (const char *filename)
- bool [ParseKeyFile](#) (const char *filename)
- void [SetCipherType](#) ([CipherTypes](#) type)
- bool [SetPassword](#) (const char *pass, size_t passLen)

Public Member Functions inherited from [gdcmm::CryptographicMessageSyntax](#)

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete

Additional Inherited Members

Public Types inherited from [gdcmm::CryptographicMessageSyntax](#)

- enum [CipherTypes](#) {
[DES3_CIPHER](#) ,
[AES128_CIPHER](#) ,
[AES192_CIPHER](#) ,
[AES256_CIPHER](#) }

12.43.1 Constructor & Destructor Documentation

12.43.1.1 CAPICryptographicMessageSyntax()

[gdcmm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax](#) ()

12.43.1.2 ~CAPICryptographicMessageSyntax()

[gdcmm::CAPICryptographicMessageSyntax::~~CAPICryptographicMessageSyntax](#) ()

12.43.2 Member Function Documentation

12.43.2.1 Decrypt()

```
bool gdcmm::CAPICryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len) const [virtual]
```

decrypt content from a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.2 Encrypt()

```
bool gdcmm::CAPICryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len) const    [virtual]
```

create a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.3 GetCipherType()

```
CipherTypes gdcmm::CAPICryptographicMessageSyntax::GetCipherType () const    [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.4 GetInitialized()

```
bool gdcmm::CAPICryptographicMessageSyntax::GetInitialized () const    [inline]
```

12.43.2.5 ParseCertificateFile()

```
bool gdcmm::CAPICryptographicMessageSyntax::ParseCertificateFile (
    const char * filename)    [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.6 ParseKeyFile()

```
bool gdcmm::CAPICryptographicMessageSyntax::ParseKeyFile (
    const char * filename)    [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.7 SetCipherType()

```
void gdcmm::CAPICryptographicMessageSyntax::SetCipherType (
    CipherTypes type)    [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.43.2.8 SetPassword()

```
bool gdcm::CAPICryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen) [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

The documentation for this class was generated from the following file:

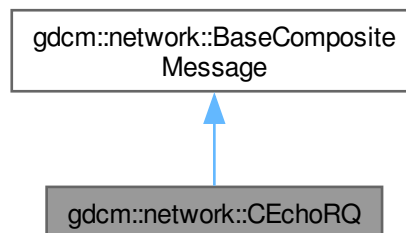
- [gdcmCAPICryptographicMessageSyntax.h](#)

12.44 gdcm::network::CEchoRQ Class Reference

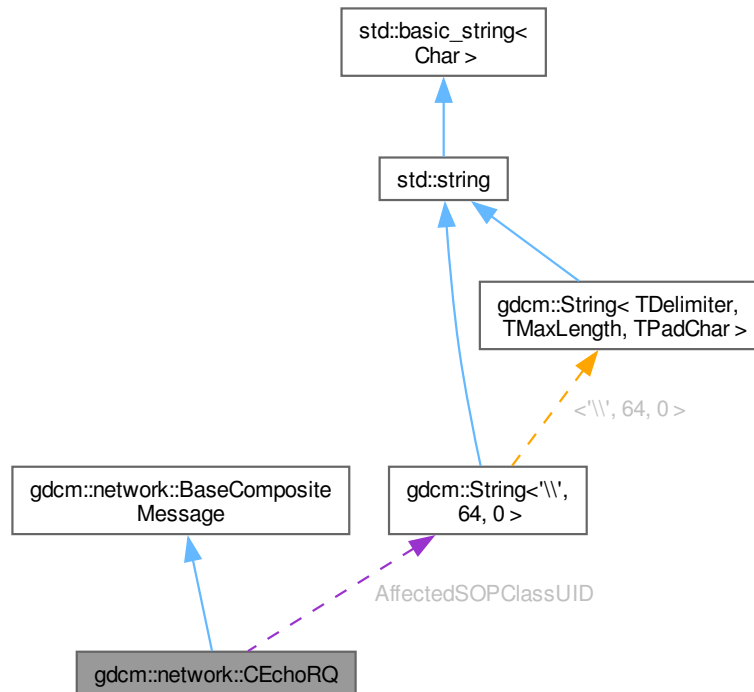
[CEchoRQ](#).

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRQ:



Collaboration diagram for `gdcm::network::CEchoRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const `ULConnection` &inConnection, const `BaseRootQuery` *inRootQuery) override

Public Member Functions inherited from `gdcm::network::BaseCompositeMessage`

- virtual `~BaseCompositeMessage` ()=default

Public Attributes

- `UIComp` `AffectedSOPClassUID`
- `uint16_t` `MessageID`

12.44.1 Detailed Description

`CEchoRQ`.

this file defines the messages for the cecho action

12.44.2 Member Function Documentation

12.44.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseCompositeMessage](#).

12.44.3 Member Data Documentation

12.44.3.1 AffectedSOPClassUID

[UIComp](#) gdcm::network::CEchoRQ::AffectedSOPClassUID

12.44.3.2 MessageID

uint16_t gdcm::network::CEchoRQ::MessageID

The documentation for this class was generated from the following files:

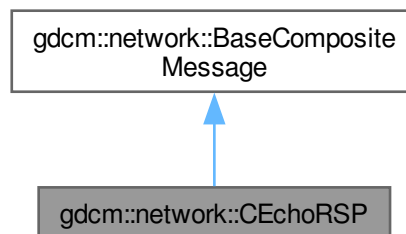
- [gdcmCEchoMessages.h](#)
- [gdcmDIMSE.h](#)

12.45 gdcm::network::CEchoRSP Class Reference

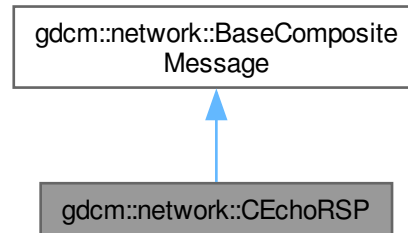
[CEchoRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRSP:



Collaboration diagram for `gdcm::network::CEchoRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseRootQuery *inRootQuery`)=0

12.45.1 Detailed Description

[CEchoRSP](#) this file defines the messages for the cecho action.

12.45.2 Member Function Documentation

12.45.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

- [gdcmCEchoMessages.h](#)

12.46 `gdcm::network::CFind` Class Reference

```
#include <gdcmDIMSE.h>
```

12.46.1 Detailed Description

PS 3.4 - 2009 [Table B.2-1 C-STORE STATUS](#)

The documentation for this class was generated from the following file:

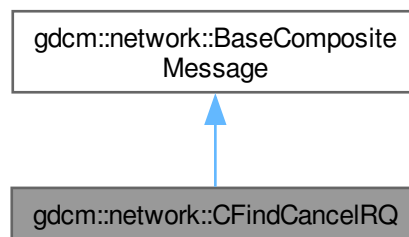
- [gdcmDIMSE.h](#)

12.47 gdcm::network::CFindCancelRQ Class Reference

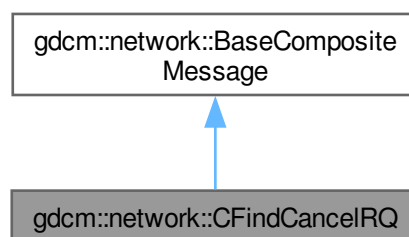
[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for `gdcm::network::CFindCancelRQ`:



Collaboration diagram for `gdcm::network::CFindCancelRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) *inDataSet)

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)=0

12.47.1 Detailed Description

[CFindCancelRQ](#) this file defines the messages for the cfind action.

12.47.2 Member Function Documentation

12.47.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

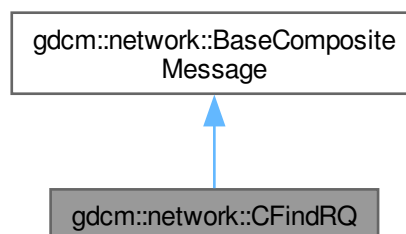
- [gdcmCFindMessages.h](#)

12.48 [gdcm::network::CFindRQ](#) Class Reference

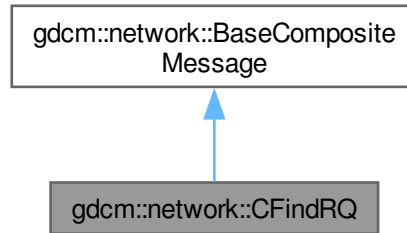
[CFindRQ](#).

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for [gdcm::network::CFindRQ](#):



Collaboration diagram for `gdcm::network::CFindRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)` override

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`

12.48.1 Detailed Description

[CFindRQ](#).

this file defines the messages for the cfind action

12.48.2 Member Function Documentation

12.48.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CFindRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseCompositeMessage](#).

The documentation for this class was generated from the following file:

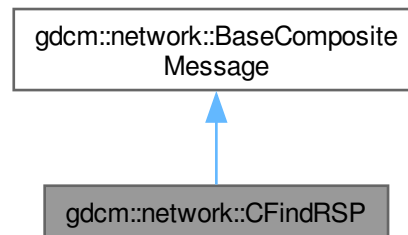
- [gdcmCFindMessages.h](#)

12.49 gdcm::network::CFindRSP Class Reference

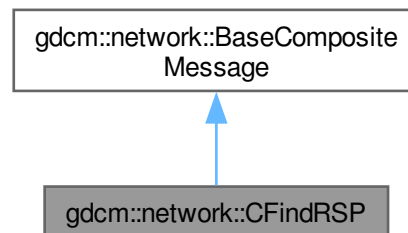
[CFindRSP](#) this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for `gdcm::network::CFindRSP`:



Collaboration diagram for `gdcm::network::CFindRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()`=default
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseRootQuery *inRootQuery`)=0

12.49.1 Detailed Description

[CFindRSP](#) this file defines the messages for the cfind action.

12.49.2 Member Function Documentation

12.49.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindRSP::ConstructPDVByDataSet (  
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

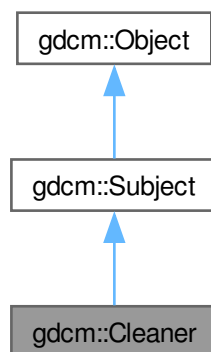
- [gdcmCFindMessages.h](#)

12.50 gdcm::Cleaner Class Reference

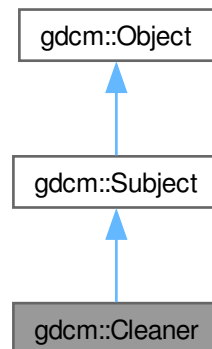
[Cleaner](#).

```
#include <gdcmCleaner.h>
```

Inheritance diagram for gdcm::Cleaner:



Collaboration diagram for `gdcm::Cleaner`:



Public Types

- typedef `std::tuple< std::string, std::string, std::string >` [CodedEntryData](#)

Public Member Functions

- [Cleaner](#) ()
- [~Cleaner](#) () override
- bool [Clean](#) ()
 - main loop
- bool [Empty](#) (DPath const &dpath)
- bool [Empty](#) (PrivateTag const &pt)
- bool [Empty](#) (Tag const &t)
- bool [Empty](#) (VR const &vr)
- void [EmptyWhenScrubFails](#) (bool empty)
 - Should I empty instead of scrub upon failure.
- [File](#) & [GetFile](#) ()
- bool [Preserve](#) (DPath const &dpath)
 - Preserve.
- bool [Remove](#) (DPath const &dpath)
- bool [Remove](#) (PrivateTag const &pt)
- bool [Remove](#) (Tag const &t)
- bool [Remove](#) (VR const &vr)
- void [RemoveAllGroupLength](#) (bool remove)
 - Should I remove all group length (deprecated). Default: true.
- void [RemoveAllIllegal](#) (bool remove)
 - Should I remove all illegal attribute. Default: true.
- void [RemoveAllMissingPrivateCreator](#) (bool remove)

- bool [RemoveMissingPrivateCreator](#) ([Tag](#) const &t)
- bool [ReplaceCodeMeaning](#) ([CodedEntryData](#) const &ced)
Coded Entry Data.
- bool [Scrub](#) ([DPath](#) const &dpath)
- bool [Scrub](#) ([PrivateTag](#) const &pt)
- bool [Scrub](#) ([Tag](#) const &t)
Clean digital tash (typically SIEMENS CSA header):
- bool [Scrub](#) ([VR](#) const &vr)
- void [SetFile](#) (const [File](#) &f)
Set/Get [File](#).

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [Cleaner](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.50.1 Detailed Description

[Cleaner](#).

This class implement the Subject/Observer pattern trigger the following event:

- [AnonymizeEvent](#)
- [IterationEvent](#)
- [StartEvent](#)
- [EndEvent](#)

Examples

[Cleaner.cs](#).

12.50.2 Member Typedef Documentation

12.50.2.1 CodedEntryData

`typedef std::tuple<std::string, std::string, std::string> gdcm::Cleaner::CodedEntryData`

12.50.3 Constructor & Destructor Documentation

12.50.3.1 Cleaner()

`gdcm::Cleaner::Cleaner ()`

Referenced by [New\(\)](#).

12.50.3.2 ~Cleaner()

`gdcm::Cleaner::~~Cleaner ()` [override]

12.50.4 Member Function Documentation

12.50.4.1 Clean()

`bool gdcm::Cleaner::Clean ()`

main loop

Examples

[Cleaner.cs](#).

12.50.4.2 Empty() [1/4]

```
bool gdcmm::Cleaner::Empty (  
    DPath const & dpath)
```

12.50.4.3 Empty() [2/4]

```
bool gdcmm::Cleaner::Empty (  
    PrivateTag const & pt)
```

12.50.4.4 Empty() [3/4]

```
bool gdcmm::Cleaner::Empty (  
    Tag const & t)
```

Examples

[Cleaner.cs](#).

12.50.4.5 Empty() [4/4]

```
bool gdcmm::Cleaner::Empty (  
    VR const & vr)
```

12.50.4.6 EmptyWhenScrubFails()

```
void gdcmm::Cleaner::EmptyWhenScrubFails (  
    bool empty)
```

Should I empty instead of scrub upon failure.

12.50.4.7 GetFile()

[File](#) & gdcmm::Cleaner::GetFile () [inline]

Examples

[Cleaner.cs](#).

12.50.4.8 New()

[SmartPointer](#)< [Cleaner](#) > gdcm::Cleaner::New () [inline], [static]

for wrapped language: instantiate a reference counted object

References [Cleaner\(\)](#).

12.50.4.9 Preserve()

bool gdcm::Cleaner::Preserve (
 [DPath](#) const & dpath)

Preserve.

Examples

[Cleaner.cs](#).

12.50.4.10 Remove() [1/4]

bool gdcm::Cleaner::Remove (
 [DPath](#) const & dpath)

12.50.4.11 Remove() [2/4]

bool gdcm::Cleaner::Remove (
 [PrivateTag](#) const & pt)

12.50.4.12 Remove() [3/4]

bool gdcm::Cleaner::Remove (
 [Tag](#) const & t)

Examples

[Cleaner.cs](#).

12.50.4.13 Remove() [4/4]

bool gdcm::Cleaner::Remove (
 [VR](#) const & vr)

12.50.4.14 RemoveAllGroupLength()

```
void gdcm::Cleaner::RemoveAllGroupLength (
    bool remove)
```

Should I remove all group length (deprecated). Default: true.

12.50.4.15 RemoveAllIllegal()

```
void gdcm::Cleaner::RemoveAllIllegal (
    bool remove)
```

Should I remove all illegal attribute. Default: true.

12.50.4.16 RemoveAllMissingPrivateCreator()

```
void gdcm::Cleaner::RemoveAllMissingPrivateCreator (
    bool remove)
```

Should I remove all private tag for which no private creator is found. Default: true

12.50.4.17 RemoveMissingPrivateCreator()

```
bool gdcm::Cleaner::RemoveMissingPrivateCreator (
    Tag const & t)
```

Specify a private tag (odd number) without a private creator (root level only for now):

12.50.4.18 ReplaceCodeMeaning()

```
bool gdcm::Cleaner::ReplaceCodeMeaning (
    CodedEntryData const & ced)
```

Coded Entry Data.

12.50.4.19 Scrub() [1/4]

```
bool gdcm::Cleaner::Scrub (
    DPath const & dpath)
```

12.50.4.20 Scrub() [2/4]

```
bool gdcm::Cleaner::Scrub (
    PrivateTag const & pt)
```

12.50.4.21 Scrub() [3/4]

```
bool gdcmm::Cleaner::Scrub (  
    Tag const & t)
```

Clean digital tash (typically SIEMENS CSA header):

Examples

[Cleaner.cs](#).

12.50.4.22 Scrub() [4/4]

```
bool gdcmm::Cleaner::Scrub (  
    VR const & vr)
```

12.50.4.23 SetFile()

```
void gdcmm::Cleaner::SetFile (  
    const File & f) [inline]
```

Set/Get [File](#).

Examples

[Cleaner.cs](#).

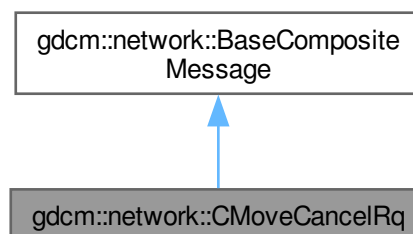
The documentation for this class was generated from the following file:

- [gdcmmCleaner.h](#)

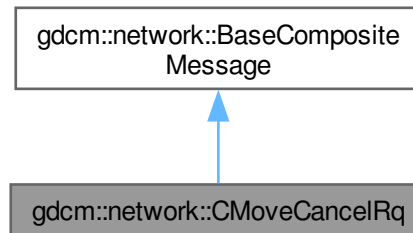
12.51 gdcmm::network::CMoveCancelRq Class Reference

```
#include <gdcmmCMoveMessages.h>
```

Inheritance diagram for gdcmm::network::CMoveCancelRq:



Collaboration diagram for gdcm::network::CMoveCancelRq:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

12.51.1 Member Function Documentation

12.51.1.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (  
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

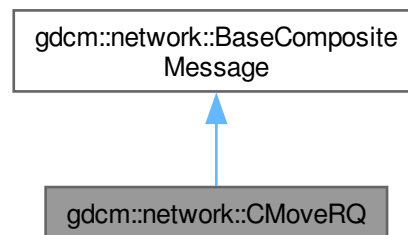
- [gdcmCMoveMessages.h](#)

12.52 gdcm::network::CMoveRQ Class Reference

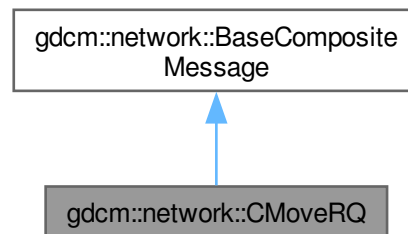
[CMoveRQ](#).

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for `gdcm::network::CMoveRQ`:



Collaboration diagram for `gdcm::network::CMoveRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery) override

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default

12.52.1 Detailed Description

[CMoveRQ](#).

this file defines the messages for the cmove action

12.52.2 Member Function Documentation

12.52.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseCompositeMessage](#).

The documentation for this class was generated from the following file:

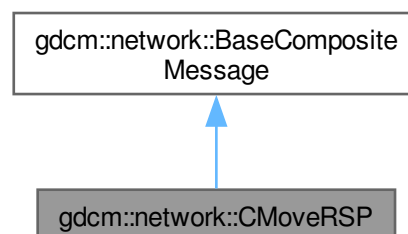
- [gdcmCMoveMessages.h](#)

12.53 gdcm::network::CMoveRSP Class Reference

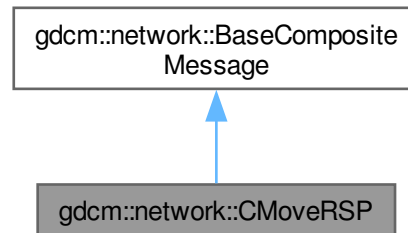
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRSP:



Collaboration diagram for `gdcm::network::CMoveRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- virtual `~BaseCompositeMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseRootQuery *inRootQuery`)=0

12.53.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

12.53.2 Member Function Documentation

12.53.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

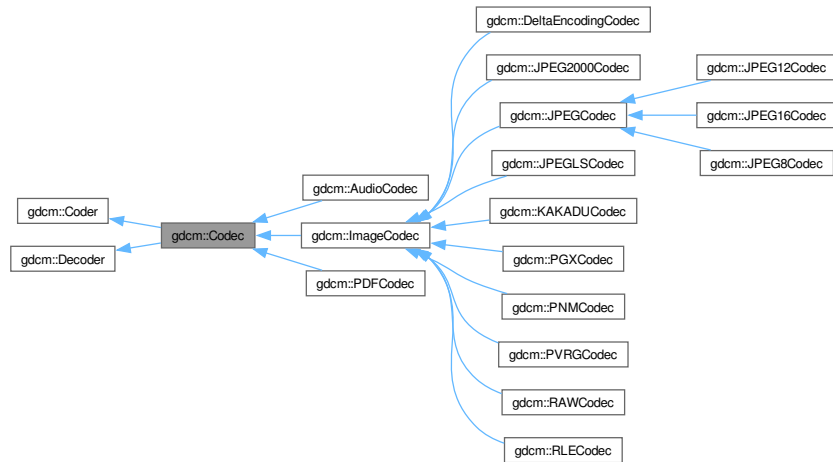
- [gdcmCMoveMessages.h](#)

12.54 gdcm::Codec Class Reference

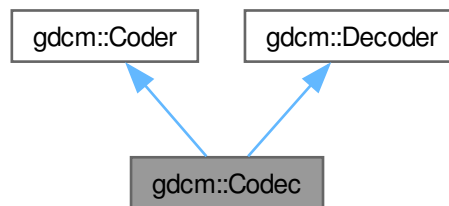
[Codec](#) class.

```
#include <gdcmCodec.h>
```

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [CanCode](#) ([TransferSyntax](#) const &) const =0
Return whether this coder support this transfer syntax (can code it).
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default
- virtual bool [CanDecode](#) ([TransferSyntax](#) const &) const =0
Return whether this decoder support this transfer syntax (can decode it).
- virtual bool [Decode](#) ([DataElement](#) const &, [DataElement](#) &)
Decode.

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from [gdcm::Decoder](#)

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

12.54.1 Detailed Description

[Codec](#) class.

The documentation for this class was generated from the following file:

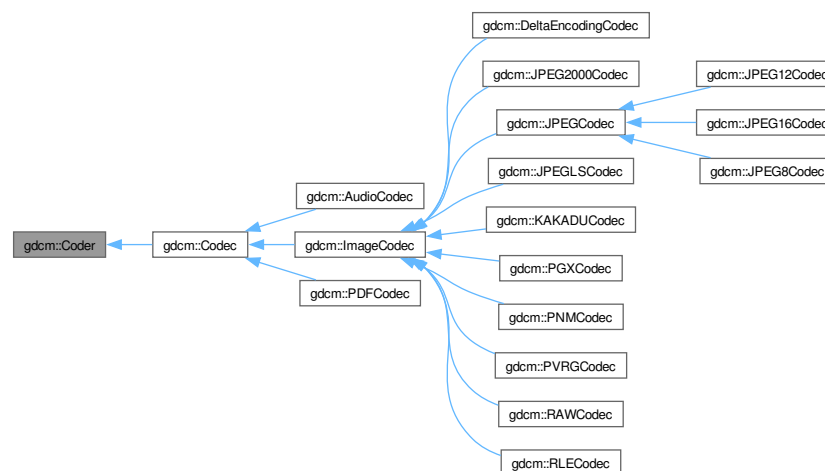
- [gdcmCodec.h](#)

12.55 gdcm::Coder Class Reference

[Coder](#).

```
#include <gdcmCoder.h>
```

Inheritance diagram for [gdcm::Coder](#):



Public Member Functions

- virtual [~Coder](#) ()=default
- virtual bool [CanCode](#) ([TransferSyntax](#) const &) const =0
Return whether this coder support this transfer syntax (can code it).
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Protected Member Functions

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

12.55.1 Detailed Description

[Coder](#).

12.55.2 Constructor & Destructor Documentation

12.55.2.1 ~Coder()

virtual gdcm::Coder::~Coder () [virtual], [default]

12.55.3 Member Function Documentation

12.55.3.1 CanCode()

virtual bool gdcm::Coder::CanCode (
[TransferSyntax](#) const &) const [pure virtual]

Return whether this coder support this transfer syntax (can code it).

Implemented in [gdcm::AudioCodec](#), [gdcm::ImageCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PDFCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

12.55.3.2 Code()

virtual bool gdcm::Coder::Code (
[DataElement](#) const & in_,
[DataElement](#) & out_) [inline], [virtual]

Code.

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

12.55.3.3 InternalCode()

```
virtual bool gdcmm::Coder::InternalCode (
    const char * bv,
    unsigned long len,
    std::ostream & os) [inline], [protected], [virtual]
```

Reimplemented in [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), and [gdcmm::JPEG8Codec](#).

The documentation for this class was generated from the following file:

- [gdcmmCoder.h](#)

12.56 gdcmm::CodeString Class Reference

[CodeString](#).

```
#include <gdcmmCodeString.h>
```

Public Types

- typedef [InternalClass::const_iterator](#) [const_iterator](#)
- typedef [InternalClass::const_reference](#) [const_reference](#)
- typedef [InternalClass::const_reverse_iterator](#) [const_reverse_iterator](#)
- typedef [InternalClass::difference_type](#) [difference_type](#)
- typedef [InternalClass::iterator](#) [iterator](#)
- typedef [InternalClass::pointer](#) [pointer](#)
- typedef [InternalClass::reference](#) [reference](#)
- typedef [InternalClass::reverse_iterator](#) [reverse_iterator](#)
- typedef [InternalClass::size_type](#) [size_type](#)
- typedef [InternalClass::value_type](#) [value_type](#)

Public Member Functions

- [CodeString](#) ()
[CodeString](#) constructors.
- [CodeString](#) (const [InternalClass](#) &s, [size_type](#) pos=0, [size_type](#) n=[InternalClass::npos](#))
- [CodeString](#) (const [value_type](#) *s)
- [CodeString](#) (const [value_type](#) *s, [size_type](#) n)
- std::string [GetAsString](#) () const
Return the full code string as std::string.
- bool [IsValid](#) () const
Check if [CodeString](#) obj is correct..
- [size_type](#) [Size](#) () const
Return the size of the string.

Protected Member Functions

- std::string [TrimInternal](#) () const

Friends

- bool [operator!=](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)
- std::ostream & [operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- bool [operator==](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)

12.56.1 Detailed Description

[CodeString](#).

This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that `operator==` is correct.

Note

the ctor of [CodeString](#) will Trim the string on the fly so as to remove the extra leading and ending spaces. However it will not perform validation on the fly ([CodeString](#) obj can contains invalid char such as lower cases). This design was chosen to be a little tolerant to broken DICOM implementation, and thus allow user to compare lower case CS from there input file without the need to first rewrite them to get rid of invalid character (validation is a different operation from searching, querying).

Warning

when writing out DICOM file it is highly recommended to perform the [IsValid\(\)](#) call, at least to check that the length of the string match the definition in the standard.

12.56.2 Member Typedef Documentation

12.56.2.1 `const_iterator`

```
typedef InternalClass::const\_iterator gdcm::CodeString::const_iterator
```

12.56.2.2 `const_reference`

```
typedef InternalClass::const\_reference gdcm::CodeString::const_reference
```

12.56.2.3 `const_reverse_iterator`

```
typedef InternalClass::const\_reverse\_iterator gdcm::CodeString::const_reverse_iterator
```

12.56.2.4 difference_type

typedef [InternalClass::difference_type](#) gdcm::CodeString::difference_type

12.56.2.5 iterator

typedef [InternalClass::iterator](#) gdcm::CodeString::iterator

12.56.2.6 pointer

typedef [InternalClass::pointer](#) gdcm::CodeString::pointer

12.56.2.7 reference

typedef [InternalClass::reference](#) gdcm::CodeString::reference

12.56.2.8 reverse_iterator

typedef [InternalClass::reverse_iterator](#) gdcm::CodeString::reverse_iterator

12.56.2.9 size_type

typedef [InternalClass::size_type](#) gdcm::CodeString::size_type

12.56.2.10 value_type

typedef [InternalClass::value_type](#) gdcm::CodeString::value_type

12.56.3 Constructor & Destructor Documentation

12.56.3.1 CodeString() [1/4]

gdcm::CodeString::CodeString () [inline]

[CodeString](#) constructors.

Referenced by [operator!=](#), [operator<<](#), and [operator==](#).

12.56.3.2 CodeString() [2/4]

```
gdcmm::CodeString::CodeString (  
    const value\_type * s) [inline]
```

12.56.3.3 CodeString() [3/4]

```
gdcmm::CodeString::CodeString (  
    const value\_type * s,  
    size\_type n) [inline]
```

12.56.3.4 CodeString() [4/4]

```
gdcmm::CodeString::CodeString (  
    const InternalClass & s,  
    size\_type pos = 0,  
    size\_type n = InternalClass::npos) [inline]
```

12.56.4 Member Function Documentation

12.56.4.1 GetAsString()

```
std::string gdcmm::CodeString::GetAsString () const [inline]
```

Return the full code string as std::string.

12.56.4.2 IsValid()

```
bool gdcmm::CodeString::IsValid () const
```

Check if [CodeString](#) obj is correct..

12.56.4.3 Size()

```
size\_type gdcmm::CodeString::Size () const [inline]
```

Return the size of the string.

12.56.4.4 TrimInternal()

```
std::string gdcmm::CodeString::TrimInternal () const [inline], [protected]
```

12.56.5 Friends And Related Symbol Documentation

12.56.5.1 operator"!=

```
bool operator!= (
    const CodeString & ref,
    const CodeString & cs) [friend]
```

References [CodeString\(\)](#).

12.56.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const CodeString & str) [friend]
```

References [CodeString\(\)](#).

12.56.5.3 operator==

```
bool operator== (
    const CodeString & ref,
    const CodeString & cs) [friend]
```

References [CodeString\(\)](#).

The documentation for this class was generated from the following file:

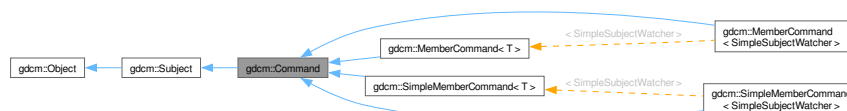
- [gdcmlCodeString.h](#)

12.57 gdcml::Command Class Reference

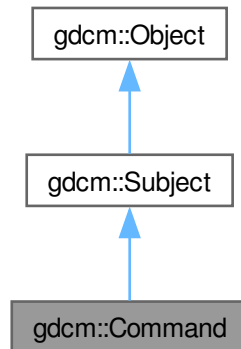
[Command](#) superclass for callback/observer methods.

```
#include <gdcmlCommand.h>
```

Inheritance diagram for gdcml::Command:



Collaboration diagram for gdcmm::Command:



Public Member Functions

- [Command](#) (const Command &)=delete
 - virtual void [Execute](#) (const [Subject](#) *caller, const [Event](#) &event)=0
 - virtual void [Execute](#) ([Subject](#) *caller, const [Event](#) &event)=0
- Abstract method that defines the action to be taken by the command.
- void [operator=](#) (const [Command](#) &)=delete

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
 - [Object](#) (const Object &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
 - void [operator=](#) (const [Object](#) &)
 - virtual void [Print](#) (std::ostream &) const

Protected Member Functions

- [Command](#) ()
- [~Command](#) () override

Protected Member Functions inherited from [gdcM::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.57.1 Detailed Description

[Command](#) superclass for callback/observer methods.

See also

[Subject](#)

12.57.2 Constructor & Destructor Documentation

12.57.2.1 [Command](#)() [1/2]

```
gdcM::Command::Command (  
    const Command & )    [delete]
```

References [Command](#)() .

Referenced by [Command](#)() , and [operator=](#)() .

12.57.2.2 [Command](#)() [2/2]

```
gdcM::Command::Command ()    [protected]
```

12.57.2.3 [~Command](#)()

```
gdcM::Command::~~Command ()    [override], [protected]
```

12.57.3 Member Function Documentation

12.57.3.1 Execute() [1/2]

```
virtual void gdcmm::Command::Execute (  
    const Subject * caller,  
    const Event & event) [pure virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const [Object](#)

Implemented in [gdcmm::MemberCommand< T >](#), [gdcmm::MemberCommand< SimpleSubjectWatcher >](#), [gdcmm::SimpleMemberCommand< T >](#), and [gdcmm::SimpleMemberCommand< SimpleSubjectWatcher >](#).

References [gdcmm::Subject::Subject\(\)](#).

12.57.3.2 Execute() [2/2]

```
virtual void gdcmm::Command::Execute (  
    Subject * caller,  
    const Event & event) [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in [gdcmm::MemberCommand< T >](#), [gdcmm::MemberCommand< SimpleSubjectWatcher >](#), [gdcmm::SimpleMemberCommand< T >](#), and [gdcmm::SimpleMemberCommand< SimpleSubjectWatcher >](#).

References [gdcmm::Subject::Subject\(\)](#).

12.57.3.3 operator=()

```
void gdcmm::Command::operator= (  
    const Command & ) [delete]
```

References [Command\(\)](#).

The documentation for this class was generated from the following file:

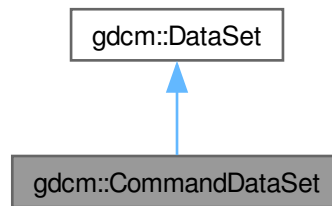
- [gdcmmCommand.h](#)

12.58 gdcm::CommandDataSet Class Reference

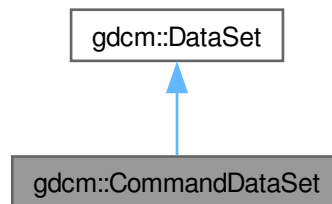
Class to represent a [Command DataSet](#).

```
#include <gdcmCommandDataSet.h>
```

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- [CommandDataSet](#) ()=default
- [~CommandDataSet](#) ()=default
- void [Insert](#) (const [DataElement](#) &de)
- std::istream & [Read](#) (std::istream &is)
Read.
- void [Replace](#) (const [DataElement](#) &de)
- std::ostream & [Write](#) (std::ostream &os) const
Write.

Public Member Functions inherited from [gdcmm::DataSet](#)

- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()
- template<typename TDE>
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const
Look up if private tag 't' is present in the dataset:
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const
Return the dataelement.
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataSet](#) & [GetDES](#) ()
- const [DataSet](#) & [GetDES](#) () const
- template<typename TDE>
[VL](#) [GetLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const
- [PrivateTag](#) [GetPrivateTag](#) (const [Tag](#) &t) const
Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsEmpty](#) () const
Returns if the dataset is empty.
- const [DataElement](#) & [operator\(\)](#) (uint16_t group, uint16_t element) const
- [DataSet](#) & [operator=](#) ([DataSet](#) const &)=default
- const [DataElement](#) & [operator\[\]](#) (const [Tag](#) &t) const
- void [Print](#) (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadNested](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTags](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTagsWithLength](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTag](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTagWithLength](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags, [VL](#) &length)

- `template<typename TDE, typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `SizeType Remove` (const `Tag` &tag)
 Completely remove a dataelement from the dataset.
- `void Replace` (const `DataElement` &de)
 Replace a dataelement with another one.
- `void ReplaceEmpty` (const `DataElement` &de)
 Only replace a DICOM attribute when it is missing or empty.
- `SizeType Size` () const
- `template<typename TDE, typename TSwap>`
`std::ostream const & Write` (std::ostream &os) const

Friends

- `std::ostream & operator<<` (std::ostream &_os, const `CommandDataSet` &_val)

Additional Inherited Members

Public Types inherited from `gdcm::DataSet`

- `typedef DataSet::const_iterator` `ConstIterator`
- `typedef std::set< DataElement >` `DataElementSet`
- `typedef DataSet::iterator` `Iterator`
- `typedef DataSet::size_type` `SizeType`

Protected Member Functions inherited from `gdcm::DataSet`

- `Tag ComputeDataElement` (const `PrivateTag` &t) const
- const `DataElement` & `GetDEEnd` () const
- `void InsertDataElement` (const `DataElement` &de)

12.58.1 Detailed Description

Class to represent a `Command DataSet`.

See also

[DataSet](#)

12.58.2 Constructor & Destructor Documentation

12.58.2.1 `CommandDataSet()`

`gdcm::CommandDataSet::CommandDataSet ()` [default]

Referenced by `~CommandDataSet()`, and `operator<<`.

12.58.2.2 ~CommandDataSet()

gdcm::CommandDataSet::~~CommandDataSet () [default]

References [CommandDataSet\(\)](#), and [operator<<](#).

12.58.3 Member Function Documentation

12.58.3.1 Insert()

void gdcm::CommandDataSet::Insert (
 const [DataElement](#) & de) [inline]

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DataElement::GetTag\(\)](#), and [gdcm::DataSet::InsertDataElement](#).

Referenced by [Replace\(\)](#).

12.58.3.2 Read()

std::istream & gdcm::CommandDataSet::Read (
 std::istream & is)

Read.

12.58.3.3 Replace()

void gdcm::CommandDataSet::Replace (
 const [DataElement](#) & de) [inline]

References [gdcm::DataElement::GetTag\(\)](#), [Insert\(\)](#), and [gdcm::DataSet::Remove\(\)](#).

12.58.3.4 Write()

std::ostream & gdcm::CommandDataSet::Write (
 std::ostream & os) const

Write.

12.58.4 Friends And Related Symbol Documentation

12.58.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const CommandDataSet & _val) [friend]
```

References [CommandDataSet\(\)](#), and [gdcM::DataSet::Print\(\)](#).

Referenced by [~CommandDataSet\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMCommandDataSet.h](#)

12.59 gdcM::network::CompositeMessageFactory Class Reference

[CompositeMessageFactory](#).

```
#include <gdcMCompositeMessageFactory.h>
```

Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructCEchoRQ](#) (const [ULConnection](#) &in↔ Connection)
- static std::vector< [PresentationDataValue](#) > [ConstructCFindRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCMoveRQ](#) (const [ULConnection](#) &in↔ Connection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRQ](#) (const [ULConnection](#) &in↔ Connection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRSP](#) (const [DataSet](#) *inDataSet, const [BasePDU](#) *inPC)

12.59.1 Detailed Description

[CompositeMessageFactory](#).

This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

12.59.2 Member Function Documentation

12.59.2.1 ConstructCEchoRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (
    const ULConnection & inConnection) [static]
```

12.59.2.2 ConstructCFindRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCFindRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery) [static]
```

12.59.2.3 ConstructCMoveRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCMoveRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery) [static]
```

12.59.2.4 ConstructCStoreRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true) [static]
```

12.59.2.5 ConstructCStoreRSP()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCStoreRSP (
    const DataSet * inDataSet,
    const BasePDU * inPC) [static]
```

The documentation for this class was generated from the following file:

- [gdcmCompositeMessageFactory.h](#)

12.60 gdcm::CompositeNetworkFunctions Class Reference

Composite Network Functions.

```
#include <gdcmCompositeNetworkFunctions.h>
```

Public Types

- typedef std::vector< [KeyValuePairType](#) > [KeyValuePairArrayType](#)
- typedef std::pair< [Tag](#), std::string > [KeyValuePairType](#)

Static Public Member Functions

- static bool [CEcho](#) (const char *remote, uint16_t portno, const char *aetitle=nullptr, const char *call=nullptr)
- static bool [CFind](#) (const char *remote, uint16_t portno, const [BaseRootQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle=nullptr, const char *call=nullptr)
- static bool [CMove](#) (const char *remote, uint16_t portno, const [BaseRootQuery](#) *query, uint16_t portscp, const char *aetitle=nullptr, const char *call=nullptr, const char *outputdir=nullptr)
- static [BaseRootQuery](#) * [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [DataSet](#) &queryds, [EQueryType](#) queryType=eFind)
- static [BaseRootQuery](#) * [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [KeyValuePairArrayType](#) &keys, [EQueryType](#) queryType=eFind)
- static bool [CStore](#) (const char *remote, uint16_t portno, const [Directory::FileNamesType](#) &filenames, const char *aetitle=nullptr, const char *call=nullptr)

12.60.1 Detailed Description

Composite Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- C-ECHO SCU
- C-FIND SCU
- C-STORE SCU
- C-MOVE SCU (+internal C-STORE SCP)

Examples

[SendFileSCU.cs](#).

12.60.2 Member Typedef Documentation

12.60.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

12.60.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

12.60.3 Member Function Documentation

12.60.3.1 CEcho()

```
bool gdcm::CompositeNetworkFunctions::CEcho (
    const char * remote,
    uint16_t portno,
    const char * aetitle = nullptr,
    const char * call = nullptr) [static]
```

The most basic network function. Use this function to ensure that the remote server is responding on the given IP and port number as expected.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

Examples

[SendFileSCU.cs](#).

12.60.3.2 CFind()

```
bool gdcm::CompositeNetworkFunctions::CFind (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle = nullptr,
    const char * call = nullptr) [static]
```

This function will use the provided query to determine what files a remote server contains that match the query strings. The return is a vector of datasets that contain tags as reported by the server. If the dataset

is empty, then it is possible that an error condition was encountered; in which case, the user should monitor the error and warning streams.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

12.60.3.3 CMove()

```
bool gdcmm::CompositeNetworkFunctions::CMove (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    uint16_t portscp,
    const char * aetitle = nullptr,
    const char * call = nullptr,
    const char * outputdir = nullptr) [static]
```

This function will use the provided query to get files from a remote server. NOTE that this functionality is essentially equivalent to C-GET in the DICOM standard; however, C-GET has been deprecated, so this function allows for the user to ask a remote server for files matching a query and return them to the local machine. Files will be written to the given output directory. If the operation succeeds, the function returns true. This function is a prime candidate for being overwritten by expert users; if the datasets should remain in memory, for instance, that behavior can be changed by creating a user-level version of this function.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
outputdir	is not set default to current dir ('.')

Returns

true if it worked.

12.60.3.4 ConstructQuery() [1/2]

```
BaseRootQuery * gdcmm::CompositeNetworkFunctions::ConstructQuery (
    ERootType inRootType,
    EQueryLevel inQueryLevel,
    const DataSet & queryds,
    EQueryType queryType = eFind) [static]
```

This function will take a list of strings and tags and fill in a query that can be used for either CFind or CMove (depending on the input boolean

Parameters

inMove).	Note that the caller is responsible for deleting the constructed query. This function is used to build both a move and a find query (true for inMove if it's move, false if it's find)
----------	--

References [gdcmm::eFind](#).

12.60.3.5 ConstructQuery() [2/2]

```
BaseRootQuery * gdcmm::CompositeNetworkFunctions::ConstructQuery (
    ERootType inRootType,
    EQueryLevel inQueryLevel,
    const KeyValuePairArrayType & keys,
    EQueryType queryType = eFind) [static]
```

Deprecated

References [gdcmm::eFind](#).

12.60.3.6 CStore()

```
bool gdcmm::CompositeNetworkFunctions::CStore (
    const char * remote,
    uint16_t portno,
    const Directory::FileNamesType & filenames,
    const char * aetitle = nullptr,
    const char * call = nullptr) [static]
```

This function will place the provided files into the remote server. The function returns true if it worked for all files.

Warning

the server side can refuse an association on a given file

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked for all files

Examples

[SendFileSCU.cs](#).

The documentation for this class was generated from the following file:

- [gdcmCompositeNetworkFunctions.h](#)

12.61 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

Public Member Functions

- [ConstCharWrapper](#) (const char *i=0)
- [operator const char *](#) () const

12.61.1 Detailed Description

Do not use me.

12.61.2 Constructor & Destructor Documentation

12.61.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (
    const char * i = 0) [inline]
```


12.61.3 Member Function Documentation

12.61.3.1 operator const char *()

gdcm::ConstCharWrapper::operator const char * () const [inline]

The documentation for this class was generated from the following file:

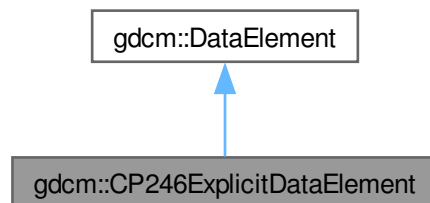
- [gdcmConstCharWrapper.h](#)

12.62 gdcm::CP246ExplicitDataElement Class Reference

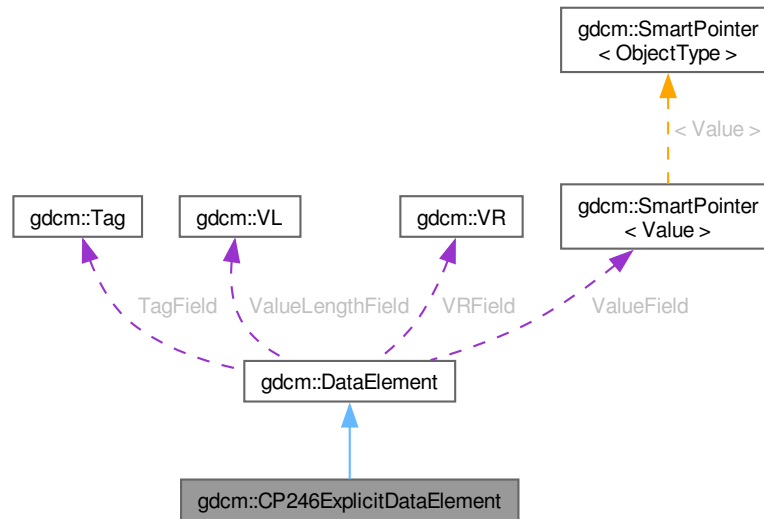
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

```
#include <gdcmCP246ExplicitDataElement.h>
```

Inheritance diagram for gdcm::CP246ExplicitDataElement:



Collaboration diagram for `gdcm::CP246ExplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear [Data Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make [Data Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const

- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.62.1 Detailed Description

Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

Note

Some system are producing SQ, declare them as UN, but encode the SQ as 'Explicit' instead of Implicit

12.62.2 Member Function Documentation

12.62.2.1 GetLength()

[VL](#) [gdcm::CP246ExplicitDataElement::GetLength](#) () const

12.62.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::Read (
    std::istream & is)
```

12.62.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.62.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```

12.62.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length)
```

The documentation for this class was generated from the following file:

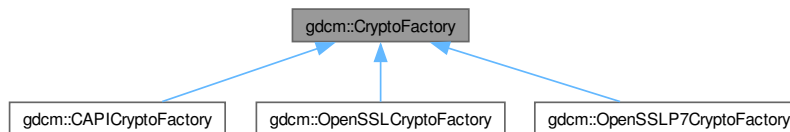
- [gdcmCP246ExplicitDataElement.h](#)

12.63 gdcm::CryptoFactory Class Reference

Class to do handle the crypto factory.

```
#include <gdcmCryptoFactory.h>
```

Inheritance diagram for gdcm::CryptoFactory:



Public Types

- enum [CryptoLib](#) {
[DEFAULT](#) = 0 ,
[OPENSSL](#) = 1 ,
[CAPI](#) = 2 ,
[OPENSSLP7](#) = 3 }

Public Member Functions

- virtual [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()=0

Static Public Member Functions

- static [CryptoFactory](#) * [GetFactoryInstance](#) ([CryptoLib](#) id=[DEFAULT](#))

Protected Member Functions

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

12.63.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independent way the user specified crypto engine. It can be:

- CAPI (windows only)
- OPENSSSL (portable)
- OPENSSLP7 (portable) By default the factory will try: CAPI if on windows OPENSSSL if possible OPENSSLP7 when older OpenSSL is used.

12.63.2 Member Enumeration Documentation

12.63.2.1 CryptoLib

enum [gdcmm::CryptoFactory::CryptoLib](#)

Enumerator

DEFAULT	
OPENSSSL	
CAPI	
OPENSSLP7	

12.63.3 Constructor & Destructor Documentation

12.63.3.1 CryptoFactory() [1/2]

[gdcmm::CryptoFactory::CryptoFactory](#) ([CryptoLib](#) id) [inline], [protected]

Referenced by [gdcmm::OpenSSLCryptoFactory::OpenSSLCryptoFactory\(\)](#), [gdcmm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory\(\)](#) and [GetFactoryInstance\(\)](#).

12.63.3.2 CryptoFactory() [2/2]

gdcm::CryptoFactory::CryptoFactory () [protected], [default]

12.63.3.3 ~CryptoFactory()

gdcm::CryptoFactory::~~CryptoFactory () [protected], [default]

12.63.4 Member Function Documentation

12.63.4.1 CreateCMSProvider()

virtual [CryptographicMessageSyntax](#) * gdcm::CryptoFactory::CreateCMSProvider () [pure virtual]

Implemented in [gdcm::CAPICryptoFactory](#), [gdcm::OpenSSLCryptoFactory](#), and [gdcm::OpenSSLP7CryptoFactory](#).

Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.63.4.2 GetFactoryInstance()

[CryptoFactory](#) * gdcm::CryptoFactory::GetFactoryInstance (
[CryptoLib](#) id = [DEFAULT](#)) [static]

References [CryptoFactory\(\)](#), and [DEFAULT](#).

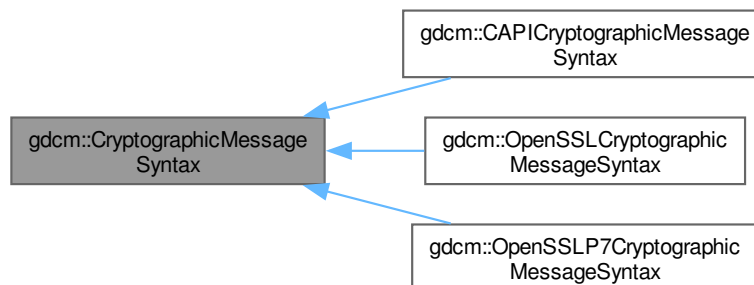
The documentation for this class was generated from the following file:

- [gdcmCryptoFactory.h](#)

12.64 gdcm::CryptographicMessageSyntax Class Reference

```
#include <gdcmCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::CryptographicMessageSyntax:



Public Types

- enum [CipherTypes](#) {
[DES3_CIPHER](#) ,
[AES128_CIPHER](#) ,
[AES192_CIPHER](#) ,
[AES256_CIPHER](#) }

Public Member Functions

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- virtual bool [Decrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const =0
 decrypt content from a CMS envelopedData structure
- virtual bool [Encrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const =0
 create a CMS envelopedData structure
- virtual [CipherTypes](#) [GetCipherType](#) () const =0
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual bool [ParseCertificateFile](#) (const char *filename)=0
- virtual bool [ParseKeyFile](#) (const char *filename)=0
- virtual void [SetCipherType](#) ([CipherTypes](#) type)=0
- virtual bool [SetPassword](#) (const char *pass, size_t passLen)=0

12.64.1 Member Enumeration Documentation

12.64.1.1 CipherTypes

enum [gdcmm::CryptographicMessageSyntax::CipherTypes](#)

Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

12.64.2 Constructor & Destructor Documentation

12.64.2.1 [CryptographicMessageSyntax](#)() [1/2]

[gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax](#) () [default]

Referenced by [CryptographicMessageSyntax\(\)](#), and [operator=\(\)](#).

12.64.2.2 ~CryptographicMessageSyntax()

virtual gdcmm::CryptographicMessageSyntax::~~CryptographicMessageSyntax () [virtual], [default]

12.64.2.3 CryptographicMessageSyntax() [2/2]

gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax (
const CryptographicMessageSyntax &) [delete]

References [CryptographicMessageSyntax\(\)](#).

12.64.3 Member Function Documentation

12.64.3.1 Decrypt()

virtual bool gdcmm::CryptographicMessageSyntax::Decrypt (
char * output,
size_t & outlen,
const char * array,
size_t len) const [pure virtual]

decrypt content from a CMS envelopedData structure

Implemented in [gdcmm::CAPICryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::OpenSSLP7CryptographicMessageSyntax](#).

12.64.3.2 Encrypt()

virtual bool gdcmm::CryptographicMessageSyntax::Encrypt (
char * output,
size_t & outlen,
const char * array,
size_t len) const [pure virtual]

create a CMS envelopedData structure

Implemented in [gdcmm::CAPICryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::OpenSSLP7CryptographicMessageSyntax](#).

12.64.3.3 GetCipherType()

virtual [CipherTypes](#) gdcmm::CryptographicMessageSyntax::GetCipherType () const [pure virtual]

Implemented in [gdcmm::CAPICryptographicMessageSyntax](#), [gdcmm::OpenSSLCryptographicMessageSyntax](#), and [gdcmm::OpenSSLP7CryptographicMessageSyntax](#).

12.64.3.4 operator=()

```
void gdcM::CryptographicMessageSyntax::operator= (  
    const CryptographicMessageSyntax & ) [delete]
```

References [CryptographicMessageSyntax\(\)](#).

12.64.3.5 ParseCertificateFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseCertificateFile (  
    const char * filename) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

Examples

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.64.3.6 ParseKeyFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseKeyFile (  
    const char * filename) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

12.64.3.7 SetCipherType()

```
virtual void gdcM::CryptographicMessageSyntax::SetCipherType (  
    CipherTypes type) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

12.64.3.8 SetPassword()

```
virtual bool gdcM::CryptographicMessageSyntax::SetPassword (  
    const char * pass,  
    size_t passLen) [pure virtual]
```

Implemented in [gdcM::CAPICryptographicMessageSyntax](#), [gdcM::OpenSSLCryptographicMessageSyntax](#), and [gdcM::OpenSSL7CryptographicMessageSyntax](#).

The documentation for this class was generated from the following file:

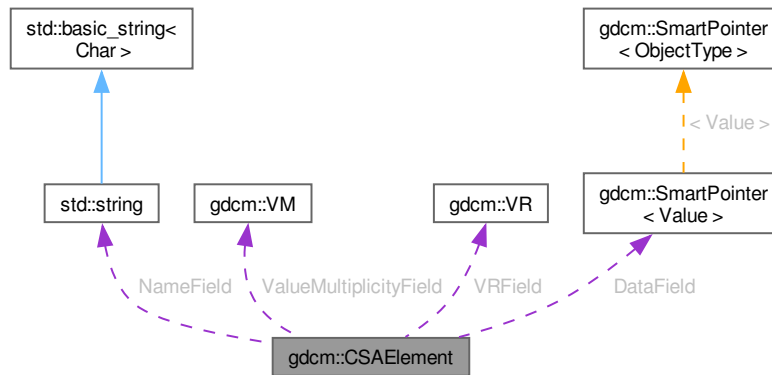
- [gdcMCryptographicMessageSyntax.h](#)

12.65 gdcm::CSAElement Class Reference

Class to represent a CSA [Element](#).

```
#include <gdcmCSAElement.h>
```

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- [CSAElement](#) (const CSAElement &_val)
- [CSAElement](#) (unsigned int kf=0)
- const [ByteValue](#) * [GetByteValue](#) () const
- unsigned int [GetKey](#) () const
Set/Get Key.
- const char * [GetName](#) () const
Set/Get Name.
- unsigned int [GetNoOffItems](#) () const
Set/Get NoOffItems.
- unsigned int [GetSyngoDT](#) () const
Set/Get SyngoDT.
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- const [VM](#) & [GetVM](#) () const
Set/Get [VM](#).
- [VR](#) const & [GetVR](#) () const
Set/Get [VR](#).
- bool [IsEmpty](#) () const
Check if CSA [Element](#) is empty.
- bool [operator<](#) (const [CSAElement](#) &de) const

- [CSAElement](#) & [operator=](#) (const [CSAElement](#) &de)=default
- bool [operator==](#) (const [CSAElement](#) &de) const
- void [SetByteValue](#) (const char *array, [VL](#) length)
Set.
- void [SetKey](#) (unsigned int key)
- void [SetName](#) (const char *name)
- void [SetNoOfItems](#) (unsigned int items)
- void [SetSyngoDT](#) (unsigned int syngodt)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVM](#) (const [VM](#) &vm)
- void [SetVR](#) ([VR](#) const &vr)

Protected Types

- typedef [SmartPointer](#)< [Value](#) > [DataPtr](#)

Protected Attributes

- [DataPtr](#) [DataField](#)
- unsigned int [KeyField](#)
- std::string [NameField](#)
- unsigned int [NoOfItemsField](#)
- unsigned int [SyngoDTField](#)
- [VM](#) [ValueMultiplicityField](#)
- [VR](#) [VRField](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [CSAElement](#) &val)

12.65.1 Detailed Description

Class to represent a CSA [Element](#).

See also

[CSAHeader](#)

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

12.65.2 Member Typedef Documentation

12.65.2.1 DataPtr

typedef [SmartPointer](#)<[Value](#)> [gdcm::CSAElement::DataPtr](#) [protected]

12.65.3 Constructor & Destructor Documentation

12.65.3.1 CSAElement() [1/2]

```
gdcmm::CSAElement::CSAElement (  
    unsigned int kf = 0) [inline]
```

References [KeyField](#), [NoOfItemsField](#), and [SyngoDTField](#).

Referenced by [CSAElement\(\)](#), [operator<\(\)](#), [operator<<](#), [operator=\(\)](#), and [operator==\(\(\)\)](#).

12.65.3.2 CSAElement() [2/2]

```
gdcmm::CSAElement::CSAElement (  
    const CSAElement & _val) [inline]
```

References [CSAElement\(\)](#).

12.65.4 Member Function Documentation

12.65.4.1 GetByteValue()

```
const ByteValue * gdcmm::CSAElement::GetByteValue () const [inline]
```

Return the [Value](#) of [CSAElement](#) as a [ByteValue](#) (if possible)

Warning

: You need to check for NULL return value

Examples

[DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

References [DataField](#).

12.65.4.2 GetKey()

```
unsigned int gdcmm::CSAElement::GetKey () const [inline]
```

Set/Get Key.

References [KeyField](#).

Referenced by [operator<\(\)](#).

12.65.4.3 GetName()

const char * gdcm::CSAElement::GetName () const [inline]

Set/Get Name.

References [NameField](#).

12.65.4.4 GetNoOfItems()

unsigned int gdcm::CSAElement::GetNoOfItems () const [inline]

Set/Get NoOfItems.

References [NoOfItemsField](#).

12.65.4.5 GetSyngoDT()

unsigned int gdcm::CSAElement::GetSyngoDT () const [inline]

Set/Get SyngoDT.

References [SyngoDTField](#).

12.65.4.6 GetValue() [1/2]

[Value](#) & gdcm::CSAElement::GetValue () [inline]

References [DataField](#).

12.65.4.7 GetValue() [2/2]

[Value](#) const & gdcm::CSAElement::GetValue () const [inline]

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples

[csa2img.cxx](#).

References [DataField](#).

12.65.4.8 GetVM()

```
const VM & gdcm::CSAElement::GetVM () const [inline]
```

Set/Get [VM](#).

References [ValueMultiplicityField](#).

12.65.4.9 GetVR()

```
VR const & gdcm::CSAElement::GetVR () const [inline]
```

Set/Get [VR](#).

References [VRField](#).

12.65.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty () const [inline]
```

Check if CSA [Element](#) is empty.

Examples

[csa2img.cxx](#).

References [DataField](#).

12.65.4.11 operator<()

```
bool gdcm::CSAElement::operator< (  
    const CSAElement & de) const [inline]
```

References [CSAElement\(\)](#), and [GetKey\(\)](#).

12.65.4.12 operator=()

```
CSAElement & gdcm::CSAElement::operator= (  
    const CSAElement & de) [default]
```

References [CSAElement\(\)](#).

12.65.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
    const CSAElement & de) const    [inline]
```

References [CSAElement\(\)](#), [KeyField](#), [NameField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

12.65.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
    const char * array,
    VL length)    [inline]
```

Set.

References [SetValue\(\)](#).

12.65.4.15 SetKey()

```
void gdcm::CSAElement::SetKey (
    unsigned int key)    [inline]
```

References [KeyField](#).

12.65.4.16 SetName()

```
void gdcm::CSAElement::SetName (
    const char * name)    [inline]
```

References [NameField](#).

12.65.4.17 SetNoOfItems()

```
void gdcm::CSAElement::SetNoOfItems (
    unsigned int items)    [inline]
```

References [NoOfItemsField](#).

12.65.4.18 SetSyngoDT()

```
void gdcm::CSAElement::SetSyngoDT (
    unsigned int syngodt)    [inline]
```

References [SyngoDTField](#).

12.65.4.19 SetValue()

```
void gdcm::CSAElement::SetValue (  
    Value const & vl) [inline]
```

References [DataField](#).

Referenced by [SetByteValue\(\)](#).

12.65.4.20 SetVM()

```
void gdcm::CSAElement::SetVM (  
    const VM & vm) [inline]
```

References [ValueMultiplicityField](#).

12.65.4.21 SetVR()

```
void gdcm::CSAElement::SetVR (  
    VR const & vr) [inline]
```

References [VRField](#).

12.65.5 Friends And Related Symbol Documentation

12.65.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const CSAElement & val) [friend]
```

References [CSAElement\(\)](#), [DataField](#), [gdcm_assert](#), [gdcm::ByteValue::GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), [KeyField](#), [NameField](#), [NoOfItemsField](#), [SyngoDTField](#), [ValueMultiplicityField](#), [gdcm::VM::VM1](#), and [VRField](#).

12.65.6 Member Data Documentation

12.65.6.1 DataField

[DataPtr](#) gdcm::CSAElement::DataField [protected]

Referenced by [GetByteValue\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [IsEmpty\(\)](#), [operator<<](#), and [SetValue\(\)](#).

12.65.6.2 KeyField

unsigned int gdcM::CSAElement::KeyField [protected]

Referenced by [CSAElement\(\)](#), [GetKey\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetKey\(\)](#).

12.65.6.3 NameField

std::string gdcM::CSAElement::NameField [protected]

Referenced by [GetName\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetName\(\)](#).

12.65.6.4 NoOfItemsField

unsigned int gdcM::CSAElement::NoOfItemsField [protected]

Referenced by [CSAElement\(\)](#), [GetNoOfItems\(\)](#), [operator<<](#), and [SetNoOfItems\(\)](#).

12.65.6.5 SyngoDTField

unsigned int gdcM::CSAElement::SyngoDTField [protected]

Referenced by [CSAElement\(\)](#), [GetSyngoDT\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetSyngoDT\(\)](#).

12.65.6.6 ValueMultiplicityField

[VM](#) gdcM::CSAElement::ValueMultiplicityField [protected]

Referenced by [GetVM\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetVM\(\)](#).

12.65.6.7 VRField

[VR](#) gdcM::CSAElement::VRField [protected]

Referenced by [GetVR\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetVR\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMCSAElement.h](#)

12.66 gdcM::CSAHeader Class Reference

Class for [CSAHeader](#).

```
#include <gdcMCSAHeader.h>
```

Public Types

- enum [CSAHeaderType](#) {
[UNKNOWN](#) = 0 ,
[SV10](#) ,
[NOMAGIC](#) ,
[DATASET_FORMAT](#) ,
[INTERFILE](#) ,
[ZEROED_OUT](#) }

Diverse format of [CSAHeader](#) as found 'in the wild'.

Public Member Functions

- [CSAHeader](#) ()
- [~CSAHeader](#) ()=default
- bool [FindCSAElementByName](#) (const char *name)
- const [CSAElement](#) & [GetCSAElementByName](#) (const char *name)
- const [DataSet](#) & [GetDataSet](#) () const
Return the [DataSet](#) output (use only if Format == DATASET_FORMAT).
- [CSAHeaderType](#) [GetFormat](#) () const
- const char * [GetInterfile](#) () const
Return the string output (use only if Format == Interfile).
- bool [GetMrProtocol](#) (const [DataSet](#) &ds, [MrProtocol](#) &mrProtocol)
- bool [LoadFromDataElement](#) ([DataElement](#) const &de)
Decode the [CSAHeader](#) from element 'de'.
- void [Print](#) (std::ostream &os) const
Print the [CSAHeader](#) (use only if Format == SV10 or NOMAGIC).

Static Public Member Functions

- static const [PrivateTag](#) & [GetCSADataInfo](#) ()
- static const [PrivateTag](#) & [GetCSAImageHeaderInfoTag](#) ()
- static const [PrivateTag](#) & [GetCSASeriesHeaderInfoTag](#) ()

Protected Member Functions

- const [CSAElement](#) & [GetCSAEEnd](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [CSAHeader](#) &d)

12.66.1 Detailed Description

Class for [CSAHeader](#).

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/NOMAGIC and DATASET_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

the API of this class might change.

[Todo](#) MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

See also

[PDBHeader](#)

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image [Module](#) in http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

12.66.2 Member Enumeration Documentation

12.66.2.1 CSAHeaderType

enum [gdcm::CSAHeader::CSAHeaderType](#)

Diverse format of [CSAHeader](#) as found 'in the wild'.

Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	

INTERFILE	
ZEROED_OUT	

12.66.3 Constructor & Destructor Documentation

12.66.3.1 CSAHeader()

gdcm::CSAHeader::CSAHeader () [inline]

References [UNKNOWN](#).

Referenced by [operator<<](#).

12.66.3.2 ~CSAHeader()

gdcm::CSAHeader::~~CSAHeader () [default]

12.66.4 Member Function Documentation

12.66.4.1 FindCSAElementByName()

```
bool gdcm::CSAHeader::FindCSAElementByName (
    const char * name)
```

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

12.66.4.2 GetCSADataInfo()

```
const PrivateTag & gdcm::CSAHeader::GetCSADataInfo () [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: [PrivateTag](#)(0x0029,0x10,"↵ SIEMENS CSA NON-IMAGE");

12.66.4.3 GetCSAEEnd()

```
const CSAElement & gdcm::CSAHeader::GetCSAEEnd () const [protected]
```

12.66.4.4 GetCSAElementByName()

```
const CSAElement & gdcm::CSAHeader::GetCSAElementByName (
    const char * name)
```

Return the [CSAElement](#) corresponding to name 'name'

Warning

Case Sensitive

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

12.66.4.5 GetCSAImageHeaderInfoTag()

```
const PrivateTag & gdcm::CSAHeader::GetCSAImageHeaderInfoTag () [static]
```

Return the private tag used by SIEMENS to store the CSA [Image](#) Header This is: [PrivateTag](#)(0x0029,0x10,"↵SIEMENS CSA HEADER");

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [PublicDict.cxx](#), and [csa2img.cxx](#).

12.66.4.6 GetCSASeriesHeaderInfoTag()

```
const PrivateTag & gdcm::CSAHeader::GetCSASeriesHeaderInfoTag () [static]
```

Return the private tag used by SIEMENS to store the CSA [Series](#) Header This is: [PrivateTag](#)(0x0029,0x20,"↵SIEMENS CSA HEADER");

Examples

[MrProtocol.cxx](#).

12.66.4.7 GetDataSet()

```
const DataSet & gdcm::CSAHeader::GetDataSet () const [inline]
```

Return the [DataSet](#) output (use only if Format == DATASET_FORMAT).

12.66.4.8 GetFormat()

[CSAHeaderType](#) gdcmm::CSAHeader::GetFormat () const

return the format of the [CSAHeader](#) SV10 and NOMAGIC are equivalent.

12.66.4.9 GetInterfile()

const char * gdcmm::CSAHeader::GetInterfile () const [inline]

Return the string output (use only if Format == Interfile).

12.66.4.10 GetMrProtocol()

bool gdcmm::CSAHeader::GetMrProtocol (
 const [DataSet](#) & ds,
 [MrProtocol](#) & mrProtocol)

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

Examples

[MrProtocol.cxx](#).

12.66.4.11 LoadFromDataElement()

bool gdcmm::CSAHeader::LoadFromDataElement (
 [DataElement](#) const & de)

Decode the [CSAHeader](#) from element 'de'.

Examples

[DumpCSA.cs](#), [DumpSiemensBase64.cxx](#), [MrProtocol.cxx](#), and [csa2img.cxx](#).

12.66.4.12 Print()

void gdcmm::CSAHeader::Print (
 std::ostream & os) const

Print the [CSAHeader](#) (use only if Format == SV10 or NOMAGIC).

Examples

[csa2img.cxx](#).

Referenced by [operator<<](#).

12.66.5 Friends And Related Symbol Documentation

12.66.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const CSAHeader & d) [friend]
```

References [CSAHeader\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmCSAHeader.h](#)

12.67 gdcm::CSAHeaderDict Class Reference

Class to represent a map of [CSAHeaderDictEntry](#).

```
#include <gdcmCSAHeaderDict.h>
```

Public Types

- typedef MapCSAHeaderDictEntry::const_iterator [ConstIterator](#)
- typedef MapCSAHeaderDictEntry::iterator [Iterator](#)
- typedef std::set< [CSAHeaderDictEntry](#) > [MapCSAHeaderDictEntry](#)

Public Member Functions

- [CSAHeaderDict](#) ()
- [CSAHeaderDict](#) (const [CSAHeaderDict](#) &_val)=delete
- void [AddCSAHeaderDictEntry](#) (const [CSAHeaderDictEntry](#) &de)
- [ConstIterator](#) [Begin](#) () const
- [ConstIterator](#) [End](#) () const
- const [CSAHeaderDictEntry](#) & [GetCSAHeaderDictEntry](#) (const char *name) const
- bool [IsEmpty](#) () const
- [CSAHeaderDict](#) & [operator=](#) (const [CSAHeaderDict](#) &_val)=delete

Protected Member Functions

- void [LoadDefault](#) ()

Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &__os, const [CSAHeaderDict](#) &_val)

12.67.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

Examples

[MrProtocol.cxx](#).

12.67.2 Member Typedef Documentation

12.67.2.1 ConstIterator

typedef MapCSAHeaderDictEntry::const_iterator [gdcm::CSAHeaderDict::ConstIterator](#)

12.67.2.2 Iterator

typedef MapCSAHeaderDictEntry::iterator [gdcm::CSAHeaderDict::Iterator](#)

12.67.2.3 MapCSAHeaderDictEntry

typedef std::set<[CSAHeaderDictEntry](#)> [gdcm::CSAHeaderDict::MapCSAHeaderDictEntry](#)

12.67.3 Constructor & Destructor Documentation

12.67.3.1 CSAHeaderDict() [1/2]

[gdcm::CSAHeaderDict::CSAHeaderDict](#) () [inline]

References [gdcm_assert](#).

Referenced by [CSAHeaderDict\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.67.3.2 CSAHeaderDict() [2/2]

[gdcm::CSAHeaderDict::CSAHeaderDict](#) (
 const [CSAHeaderDict](#) & __val) [delete]

References [CSAHeaderDict\(\)](#), and [operator<<](#).

12.67.4 Member Function Documentation

12.67.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (  
    const CSAHeaderDictEntry & de) [inline]
```

References [gdcm__assert](#).

12.67.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin () const [inline]
```

12.67.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End () const [inline]
```

12.67.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry & gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (  
    const char * name) const [inline]
```

Examples

[MrProtocol.cxx](#).

12.67.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty () const [inline]
```

12.67.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault () [protected]
```

12.67.4.7 operator=()

```
CSAHeaderDict & gdcm::CSAHeaderDict::operator= (  
    const CSAHeaderDict & __val) [delete]
```

References [CSAHeaderDict\(\)](#).

12.67.5 Friends And Related Symbol Documentation

12.67.5.1 Dicts

friend class Dicts [friend]

References [Dicts](#).

Referenced by [Dicts](#).

12.67.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const CSAHeaderDict & __val) [friend]
```

References [CSAHeaderDict\(\)](#).

Referenced by [CSAHeaderDict\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDict.h](#)

12.68 gdcm::CSAHeaderDictEntry Class Reference

Class to represent an Entry in the [Dict](#).

```
#include <gdcmCSAHeaderDictEntry.h>
```

Public Member Functions

- [CSAHeaderDictEntry](#) (const char *name="", [VR](#) const &vr=[VR::INVALID](#), [VM](#) const &vm=[VM::VM0](#), const char *desc="")
- const char * [GetDescription](#) () const
Set/Get Description.
- const char * [GetName](#) () const
Set/Get Name.
- const [VM](#) & [GetVM](#) () const
Set/Get [VM](#).
- const [VR](#) & [GetVR](#) () const
Set/Get [VR](#).
- bool [operator<](#) (const [CSAHeaderDictEntry](#) &zentry) const
- void [SetDescription](#) (const char *desc)
- void [SetName](#) (const char *name)
- void [SetVM](#) ([VM](#) const &vm)
- void [SetVR](#) (const [VR](#) &vr)

Friends

- `std::ostream & operator<< (std::ostream &_os, const CSAHeaderDictEntry &_val)`

12.68.1 Detailed Description

Class to represent an Entry in the [Dict](#).

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdcm::Tag](#) to the needed information

Note

bla TODO FIXME: Need a PublicCSAHeaderDictEntry...indeed [CSAHeaderDictEntry](#) has a notion of retired which does not exist in PrivateCSAHeaderDictEntry...

See also

[gdcm::Dict](#)

Examples

[MrProtocol.cxx](#).

12.68.2 Constructor & Destructor Documentation

12.68.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
    const char * name = "",
    VR const & vr = VR::INVALID,
    VM const & vm = VM::VM0,
    const char * desc = "") [inline]
```

References [gdcm::VR::INVALID](#), and [gdcm::VM::VM0](#).

Referenced by [operator<\(\)](#), and [operator<<](#).

12.68.3 Member Function Documentation

12.68.3.1 GetDescription()

```
const char * gdcm::CSAHeaderDictEntry::GetDescription () const [inline]
```

Set/Get Description.

12.68.3.2 GetName()

```
const char * gdcmm::CSAHeaderDictEntry::GetName () const [inline]
```

Set/Get Name.

Referenced by [operator<\(\)](#).

12.68.3.3 GetVM()

```
const VM & gdcmm::CSAHeaderDictEntry::GetVM () const [inline]
```

Set/Get [VM](#).

12.68.3.4 GetVR()

```
const VR & gdcmm::CSAHeaderDictEntry::GetVR () const [inline]
```

Set/Get [VR](#).

12.68.3.5 operator<()

```
bool gdcmm::CSAHeaderDictEntry::operator< (  
    const CSAHeaderDictEntry & entry) const [inline]
```

References [CSAHeaderDictEntry\(\)](#), and [GetName\(\)](#).

12.68.3.6 setDescription()

```
void gdcmm::CSAHeaderDictEntry::setDescription (  
    const char * desc) [inline]
```

12.68.3.7 setName()

```
void gdcmm::CSAHeaderDictEntry::setName (  
    const char * name) [inline]
```

12.68.3.8 setVM()

```
void gdcmm::CSAHeaderDictEntry::setVM (  
    VM const & vm) [inline]
```

12.68.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
    const VR & vr) [inline]
```

12.68.4 Friends And Related Symbol Documentation

12.68.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const CSAHeaderDictEntry & _val) [friend]
```

References [CSAHeaderDictEntry\(\)](#).

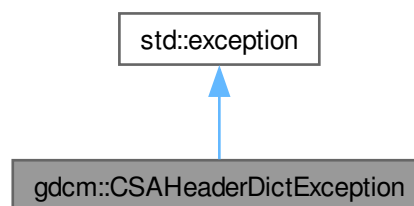
The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

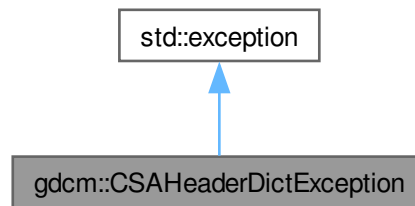
12.69 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



The documentation for this class was generated from the following file:

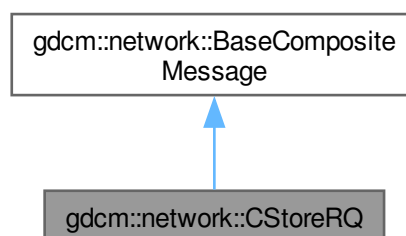
- [gdcmCSAHeaderDict.h](#)

12.70 gdcm::network::CStoreRQ Class Reference

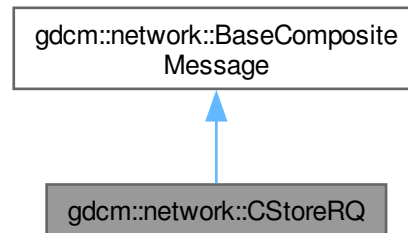
[CStoreRQ](#).

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRQ:



Collaboration diagram for `gdcm::network::CStoreRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const File &file`, `bool writeDataSet=true`)

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`

12.70.1 Detailed Description

[CStoreRQ](#).

this file defines the messages for the cecho action

12.70.2 Member Function Documentation

12.70.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRQ::ConstructPDV (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true)
```

The documentation for this class was generated from the following file:

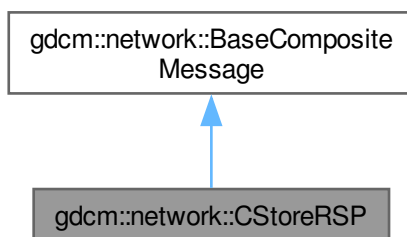
- [gdcmCStoreMessages.h](#)

12.71 gdcm::network::CStoreRSP Class Reference

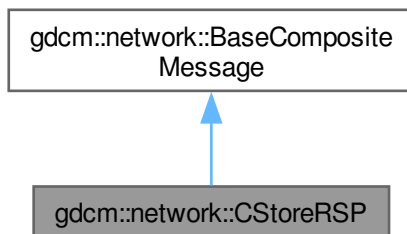
[CStoreRSP](#) this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRSP:



Collaboration diagram for gdcm::network::CStoreRSP:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV (const DataSet *inDataSet, const BasePDU *inPC)`

Public Member Functions inherited from [gdcm::network::BaseCompositeMessage](#)

- `virtual ~BaseCompositeMessage ()=default`

12.71.1 Detailed Description

[CStoreRSP](#) this file defines the messages for the cecho action.

12.71.2 Member Function Documentation

12.71.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::CStoreRSP::ConstructPDV (  
    const DataSet * inDataSet,  
    const BasePDU * inPC)
```

The documentation for this class was generated from the following file:

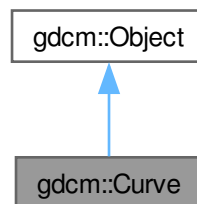
- [gdcmmCStoreMessages.h](#)

12.72 gdcmm::Curve Class Reference

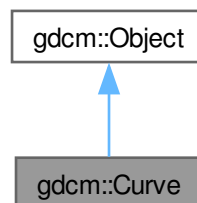
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

```
#include <gdcmmCurve.h>
```

Inheritance diagram for gdcmm::Curve:



Collaboration diagram for gdcmm::Curve:



Public Member Functions

- [Curve](#) ()
- [Curve](#) (Curve const &ov)
- [~Curve](#) () override
- void [Decode](#) (std::istream &is, std::ostream &os)
- void [GetAsPoints](#) (float *array) const
- std::vector< unsigned short > const & [GetCurveDataDescriptor](#) () const
- unsigned short [GetDataValueRepresentation](#) () const
- unsigned short [GetDimensions](#) () const
- unsigned short [GetGroup](#) () const
- unsigned short [GetNumberOfPoints](#) () const
- const char * [GetTypeOfData](#) () const
- const char * [GetTypeOfDataDescription](#) () const
- bool [IsEmpty](#) () const
- void [Print](#) (std::ostream &) const override
- void [SetCoordinateStartValue](#) (unsigned short v)
- void [SetCoordinateStepValue](#) (unsigned short v)
- void [SetCurve](#) (const char *array, unsigned int length)
- void [SetCurveDataDescriptor](#) (const uint16_t *values, size_t num)
- void [SetCurveDescription](#) (const char *curvedescription)
- void [SetDataValueRepresentation](#) (unsigned short datavaluerepresentation)
- void [SetDimensions](#) (unsigned short dimensions)
- void [SetGroup](#) (unsigned short group)
- void [SetNumberOfPoints](#) (unsigned short numberofpoints)
- void [SetTypeOfData](#) (const char *typeofdata)
- void [Update](#) (const [DataElement](#) &de)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const Object &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static unsigned int [GetNumberOfCurves](#) ([DataSet](#) const &ds)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.72.1 Detailed Description

[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

WARNING: This is deprecated and lastly defined in PS 3.3 - 2004

Examples:

- GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
- GE_DLX-8-MONO2-Multiframe.dcm
- gdcmSampleData/Philips_Medical_Images/integris_HV_5000/xa_integris.dcm
- TOSHIBA-CurveData[1-3].dcm

12.72.2 Constructor & Destructor Documentation

12.72.2.1 [Curve\(\)](#) [1/2]

```
gdcm::Curve::Curve ()
```

Referenced by [Curve\(\)](#).

12.72.2.2 [~Curve\(\)](#)

```
gdcm::Curve::~~Curve () [override]
```

12.72.2.3 [Curve\(\)](#) [2/2]

```
gdcm::Curve::Curve (  
    Curve const & ov)
```

References [Curve\(\)](#).

12.72.3 Member Function Documentation

12.72.3.1 [Decode\(\)](#)

```
void gdcm::Curve::Decode (  
    std::istream & is,  
    std::ostream & os)
```

12.72.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
    float * array) const
```

12.72.3.3 GetCurveDataDescriptor()

```
std::vector< unsigned short > const & gdcm::Curve::GetCurveDataDescriptor () const
```

12.72.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation () const
```

12.72.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions () const
```

12.72.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup () const
```

12.72.3.7 GetNumberOfCurves()

```
unsigned int gdcm::Curve::GetNumberOfCurves (
    DataSet const & ds) [static]
```

12.72.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints () const
```

12.72.3.9 GetTypeOfData()

```
const char * gdcm::Curve::GetTypeOfData () const
```

12.72.3.10 GetTypeOfDataDescription()

```
const char * gdcm::Curve::GetTypeOfDataDescription () const
```

12.72.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty () const
```

12.72.3.12 Print()

```
void gdcm::Curve::Print (  
    std::ostream & ) const    [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

12.72.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (  
    unsigned short v)
```

12.72.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (  
    unsigned short v)
```

12.72.3.15 SetCurve()

```
void gdcm::Curve::SetCurve (  
    const char * array,  
    unsigned int length)
```

12.72.3.16 SetCurveDataDescriptor()

```
void gdcm::Curve::SetCurveDataDescriptor (  
    const uint16_t * values,  
    size_t num)
```

12.72.3.17 SetCurveDescription()

```
void gdcm::Curve::SetCurveDescription (  
    const char * curvedescription)
```

12.72.3.18 SetDataValueRepresentation()

```
void gdcm::Curve::SetDataValueRepresentation (  
    unsigned short datavaluerepresentation)
```

12.72.3.19 SetDimensions()

```
void gdcm::Curve::SetDimensions (
    unsigned short dimensions)
```

12.72.3.20 SetGroup()

```
void gdcm::Curve::SetGroup (
    unsigned short group)
```

12.72.3.21 SetNumberOfPoints()

```
void gdcm::Curve::SetNumberOfPoints (
    unsigned short numberOfpoints)
```

12.72.3.22 SetTypeOfData()

```
void gdcm::Curve::SetTypeOfData (
    const char * typeofdata)
```

12.72.3.23 Update()

```
void gdcm::Curve::Update (
    const DataElement & de)
```

The documentation for this class was generated from the following file:

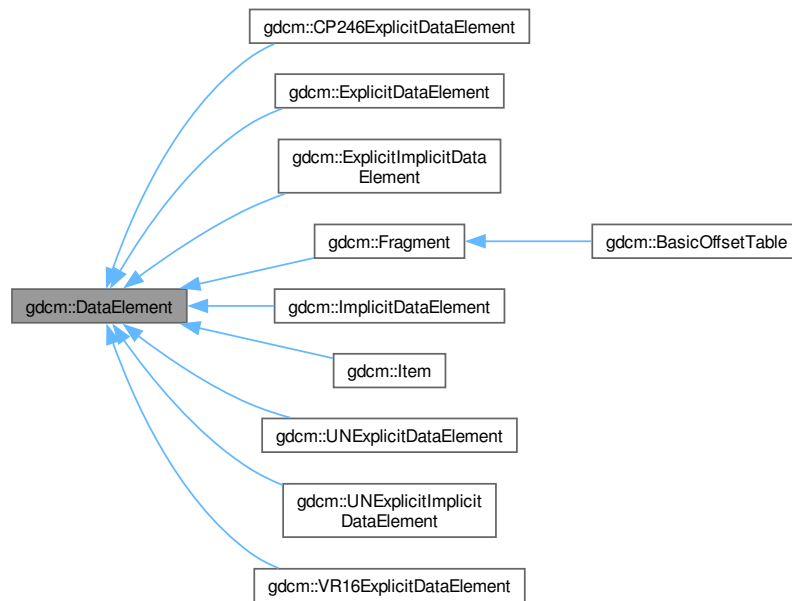
- [gdcmCurve.h](#)

12.73 gdcm::DataElement Class Reference

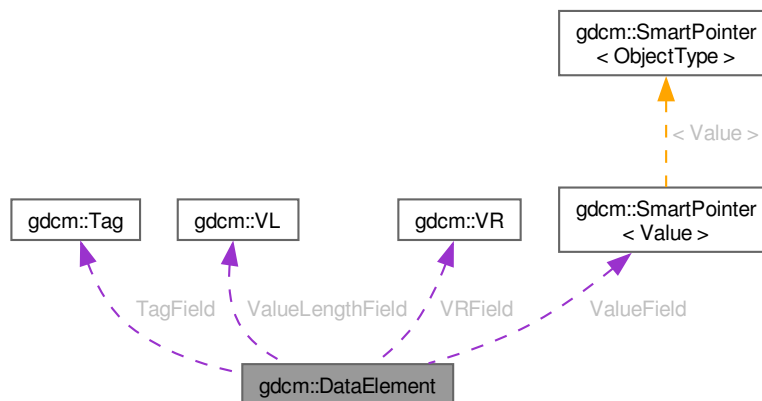
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for `gdcm::DataElement`:



Collaboration diagram for `gdcm::DataElement`:



Public Member Functions

- [DataElement](#) (const `DataElement` &_val)

- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
 - Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
 - Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
 - [VL](#) [GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
 - Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
 - Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
 - Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
 - Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
 - return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
 - std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
 - std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 - std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 - std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 - std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
 - std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
 - const std::ostream & [Write](#) (std::ostream &os) const

Protected Types

- typedef [SmartPointer](#) < [Value](#) > [ValuePtr](#)

Protected Member Functions

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [DataElement](#) &_val)

12.73.1 Detailed Description

Class to represent a Data [Element](#) either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information [Object](#) Definition (IOD) [Attribute](#) that is composed of, at a minimum, three fields: a Data [Element](#) [Tag](#), a [Value](#) Length, and a [Value](#) Field. For some specific Transfer Syntaxes, a Data [Element](#) also contains a [VR](#) Field where the [Value](#) Representation of that Data [Element](#) is specified explicitly.

Design:

- A [DataElement](#) in GDCM always store [VL](#) ([Value](#) Length) on a 32 bits integer even when [VL](#) is 16 bits
- A [DataElement](#) always store the [VR](#) even for Implicit TS, in which case [VR](#) is defaulted to [VR::INVALID](#)
- For [Item](#) start/end (See 0xfffe tags), [Value](#) is NULL

See also

[ExplicitDataElement](#) [ImplicitDataElement](#)

Examples

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [StreamImageReaderTest.cxx](#), [csa2img.cxx](#), [gdcmrtnplan.cxx](#), [gdcmrtpplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.73.2 Member Typedef Documentation

12.73.2.1 ValuePtr

typedef [SmartPointer<Value>](#) [gdcm::DataElement::ValuePtr](#) [protected]

12.73.3 Constructor & Destructor Documentation

12.73.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
    const Tag & t = Tag(0),
    const VL & vl = 0,
    const VR & vr = VR::INVALID) [inline]
```

References [gdcm::VR::INVALID](#), [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

Referenced by [DataElement\(\)](#), [gdcm::Fragment::Fragment\(\)](#), [gdcm::Item::Item\(\)](#), [gdcm::Item::Item\(\)](#), [gdcm::Item::GetDataElement\(\)](#), [gdcm::Item::InsertDataElement\(\)](#), [operator<\(\)](#), [operator<<](#), [operator=\(\)](#), and [operator==\(\)](#).

12.73.3.2 DataElement() [2/2]

```
gdcm::DataElement::DataElement (
    const DataElement & _val) [inline]
```

References [DataElement\(\)](#).

12.73.4 Member Function Documentation

12.73.4.1 Clear()

```
void gdcm::DataElement::Clear () [inline]
```

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).

References [gdcm::VR::INVALID](#), [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

Referenced by [gdcm::Item::Clear\(\)](#).

12.73.4.2 Empty()

```
void gdcm::DataElement::Empty () [inline]
```

Make Data [Element](#) empty (no [Value](#)).

References [ValueField](#), and [ValueLengthField](#).

12.73.4.3 GetByteValue()

const [ByteValue](#) * gdcmm::DataElement::GetByteValue () const [inline]

Return the [Value](#) of [DataElement](#) as a [ByteValue](#) (if possible)

Warning

: You need to check for NULL return value

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [ValueField](#).

Referenced by [IsEmpty\(\)](#), [gdcmm::BasicOffsetTable::operator<<](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::SetFrom](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), [gdcmm::Element< TVR, TVM >::SetFromDataElement\(\)](#), [gdcmm::Element< TVR, VM::VM1_n >::SetFromDataElement\(\)](#), and [gdcmm::Fragment::Write\(\)](#).

12.73.4.4 GetLength()

template<typename TDE>
[VL](#) gdcmm::DataElement::GetLength () const [inline]

References [GetLength\(\)](#).

Referenced by [GetLength\(\)](#).

12.73.4.5 GetSequenceOfFragments() [1/2]

[SequenceOfFragments](#) * gdcmm::DataElement::GetSequenceOfFragments ()

12.73.4.6 GetSequenceOfFragments() [2/2]

const [SequenceOfFragments](#) * gdcmm::DataElement::GetSequenceOfFragments () const

Return the [Value](#) of [DataElement](#) as a Sequence Of Fragments (if possible)

Warning

: You need to check for NULL return value

Examples

[DecompressImage.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

12.73.4.7 GetTag() [1/2]

[Tag](#) & gdcm::DataElement::GetTag () [inline]

References [TagField](#).

12.73.4.8 GetTag() [2/2]

const [Tag](#) & gdcm::DataElement::GetTag () const [inline]

Get [Tag](#).

Examples

[DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [SimplePrint.cs](#), and [pmsct_rgb1.cxx](#).

References [TagField](#).

Referenced by [gdcm::CommandDataSet::Insert\(\)](#), [gdcm::DataSet::Insert\(\)](#), [gdcm::FileMetaInformation::Insert\(\)](#), [operator<\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::CommandDataSet::Replace\(\)](#), [gdcm::FileMetaInformation::Replace\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#).

12.73.4.9 GetValue() [1/2]

[Value](#) & gdcm::DataElement::GetValue () [inline]

References [gdcmAssertAlwaysMacro](#), and [ValueField](#).

12.73.4.10 GetValue() [2/2]

[Value](#) const & gdcm::DataElement::GetValue () const [inline]

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples

[ReadAndDumpDICOMDIR.cxx](#).

References [gdcmAssertAlwaysMacro](#), and [ValueField](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::Element< TVR, TVM >::SetFromDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement\(\)](#).

12.73.4.11 GetValueAsSQ()

[SmartPointer](#)< [SequenceOfItems](#) > gdcm::DataElement::GetValueAsSQ () const

Interpret the [Value](#) stored in the [DataElement](#). This is more robust (but also more expensive) to call this function rather than the simplest form: `GetSequenceOfItems()` It also return NULL when the [Value](#) is NOT of type [SequenceOfItems](#)

Warning

in case `GetSequenceOfItems()` succeed the function return this value, otherwise it creates a new [SequenceOfItems](#), you should handle that in your case, for instance: [SmartPointer<SequenceOfItems>](#)
`sqi = de.GetValueAsSQ();`

Examples

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#),
[DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#),
[GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#),
[SimplePrint.cs](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

12.73.4.12 GetVL() [1/2]

[VL](#) & gdcm::DataElement::GetVL () [inline]

References [ValueLengthField](#).

12.73.4.13 GetVL() [2/2]

const [VL](#) & gdcm::DataElement::GetVL () const [inline]

Get [VL](#).

Examples

[SimplePrint.cs](#).

References [ValueLengthField](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), and [gdcm::SequenceOfFragments::Read](#)

12.73.4.14 GetVR()

[VR](#) const & gdcm::DataElement::GetVR () const [inline]

Get [VR](#) do not set [VR::SQ](#) on bytevalue data element

Examples

[DuplicatePCDE.cxx](#), and [GenFakeIdentifyFile.cxx](#).

References [VRField](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement\(\), gdcm::Element< TVR, TVM >::GetAsDataElement\(\), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement\(\), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement\(\), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement\(\)](#).

12.73.4.15 IsEmpty()

bool gdcm::DataElement::IsEmpty () const [inline]

Check if Data [Element](#) is empty.

Examples

[DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAIBugJPEGLS.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [GetByteValue\(\)](#), and [ValueField](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement\(\)](#), and [gdcm::Fragment::Write\(\)](#).

12.73.4.16 IsUndefinedLength()

bool gdcm::DataElement::IsUndefinedLength () const [inline]

return if [Value](#) Length if of undefined length

References [ValueLengthField](#).

Referenced by [gdcm::Item::InsertDataElement\(\)](#).

12.73.4.17 operator<()

```
bool gdcmm::DataElement::operator< (  
    const DataElement & de) const    [inline]
```

References [DataElement\(\)](#), and [GetTag\(\)](#).

12.73.4.18 operator=()

```
DataElement & gdcmm::DataElement::operator= (  
    const DataElement & )    [default]
```

References [DataElement\(\)](#).

12.73.4.19 operator==(())

```
bool gdcmm::DataElement::operator==( (  
    const DataElement & de) const    [inline]
```

References [DataElement\(\)](#), [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

12.73.4.20 Read()

```
template<typename TDE, typename TSwap>  
std::istream & gdcmm::DataElement::Read (  
    std::istream & is)    [inline]
```

Examples

[DumpSiemensBase64.cxx](#).

References [Read\(\)](#).

Referenced by [Read\(\)](#), and [ReadOrSkip\(\)](#).

12.73.4.21 ReadOrSkip()

```
template<typename TDE, typename TSwap>  
std::istream & gdcmm::DataElement::ReadOrSkip (  
    std::istream & is,  
    std::set< Tag > const & skiptags)    [inline]
```

References [Read\(\)](#).

12.73.4.22 ReadPreValue()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataElement::ReadPreValue (
    std::istream & is,
    std::set< Tag > const & skiptags) [inline]
```

References [ReadPreValue\(\)](#).

Referenced by [ReadPreValue\(\)](#).

12.73.4.23 ReadValue()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataElement::ReadValue (
    std::istream & is,
    std::set< Tag > const & skiptags) [inline]
```

References [ReadValue\(\)](#).

Referenced by [ReadValue\(\)](#).

12.73.4.24 ReadValueWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    std::set< Tag > const & skiptags) [inline]
```

References [ReadValueWithLength\(\)](#).

Referenced by [ReadValueWithLength\(\)](#).

12.73.4.25 ReadWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataElement::ReadWithLength (
    std::istream & is,
    VL & length) [inline]
```

References [ReadWithLength\(\)](#).

Referenced by [gdcmm::Item::Read\(\)](#), and [ReadWithLength\(\)](#).

12.73.4.26 SetByteValue()

```
void gdcm::DataElement::SetByteValue (
    const char * array,
    VL length) [inline]
```

Set the byte value

Warning

user need to read DICOM standard for an understanding of:

- even padding
- `\0` vs space padding By default even padding is achieved using `\0` regardless of the of [VR](#)

Examples

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), [iU22tomultisc.cxx](#), and [rle2img.cxx](#).

References [SetValue\(\)](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement\(\)](#), [gdcm::Element< TVR, TVM >::GetAsDataElement\(\)](#) and [gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement\(\)](#).

12.73.4.27 SetTag()

```
void gdcm::DataElement::SetTag (
    const Tag & t) [inline]
```

Set [Tag](#) Use with cautious (need to match Part 6)

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenFakeIdentifyFile.cxx](#), and [GetSubSequenceData.cxx](#).

References [TagField](#).

12.73.4.28 SetValue()

```
void gdcmm::DataElement::SetValue (
    Value const & vl) [inline]
```

Warning

you need to set the ValueLengthField explicitly

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DuplicatePCDE.cxx](#), [Fake_Image_Using_Stream_Image_V](#)
[FixBrokenJ2K.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [MpegVideoInfo.cs](#), and
[NewSequence.cs](#).

References [gdcmm::Value::GetLength\(\)](#), [ValueField](#), and [ValueLengthField](#).

Referenced by [SetByteValue\(\)](#).

12.73.4.29 SetValueFieldLength()

```
void gdcmm::DataElement::SetValueFieldLength (
    VL vl,
    bool readvalues) [protected]
```

12.73.4.30 SetVL()

```
void gdcmm::DataElement::SetVL (
    const VL & vl) [inline]
```

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

See also

[SetByteValue](#)

References [ValueLengthField](#).

12.73.4.31 SetVLToUndefined()

```
void gdcmm::DataElement::SetVLToUndefined ()
```

Examples

[Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#),
[GenSeqs.cxx](#), and [NewSequence.cs](#).

12.73.4.32 SetVR()

```
void gdcM::DataElement::SetVR (
    VR const & vr) [inline]
```

Set [VR](#) Use with cautious (need to match Part 6), advanced user only

Precondition

vr is a [VR::VRALL](#) (not a dual one such as OB_OW)

Examples

[Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#),
[GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [NewSequence.cs](#),
[StreamImageReaderTest.cxx](#), [iU2tomultisc.cxx](#), and [rle2img.cxx](#).

References [gdcM::VR::IsVRFile\(\)](#), and [VRField](#).

Referenced by [gdcM::Attribute< Group, Element, TVR, TVM >::GetAsDataElement\(\)](#), [gdcM::Attribute< Group, Element, gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement\(\)](#), [gdcM::Element< TVR, TVM >::GetAsDataElement\(\)](#), and [gdcM::Element< TVR, VM::VM1_n >::GetAsDataElement\(\)](#).

12.73.4.33 Write()

```
template<typename TDE, typename TSwap>
const std::ostream & gdcM::DataElement::Write (
    std::ostream & os) const [inline]
```

References [Write\(\)](#).

Referenced by [Write\(\)](#).

12.73.5 Friends And Related Symbol Documentation

12.73.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const DataElement & __val) [friend]
```

References [DataElement\(\)](#), [operator<<](#), [gdcM::Object::Print\(\)](#), [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

Referenced by [operator<<](#).

12.73.6 Member Data Documentation

12.73.6.1 TagField

[Tag](#) gdcm::DataElement::TagField [protected]

Referenced by [DataElement\(\)](#), [Clear\(\)](#), [GetTag\(\)](#), [GetTag\(\)](#), [operator<<](#), [gdcm::Fragment::operator<<](#), [gdcm::Item::operator<<](#), [operator==\(\)](#), [gdcm::BasicOffsetTable::Read\(\)](#), [gdcm::Item::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), [gdcm::Fragment::ReadPreValue\(\)](#), [SetTag\(\)](#), [gdcm::Fragment::Write\(\)](#), and [gdcm::Item::Write\(\)](#).

12.73.6.2 ValueField

[ValuePtr](#) gdcm::DataElement::ValueField [protected]

Referenced by [DataElement\(\)](#), [Clear\(\)](#), [Empty\(\)](#), [GetByteValue\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [IsEmpty\(\)](#), [gdcm::BasicOffsetTable::operator<<](#), [operator<<](#), [gdcm::Fragment::operator<<](#), [operator==\(\)](#), [gdcm::BasicOffsetTable::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), [gdcm::Fragment::ReadValue\(\)](#), and [SetValue\(\)](#).

12.73.6.3 ValueLengthField

[VL](#) gdcm::DataElement::ValueLengthField [protected]

Referenced by [DataElement\(\)](#), [Clear\(\)](#), [Empty\(\)](#), [GetVL\(\)](#), [GetVL\(\)](#), [IsUndefinedLength\(\)](#), [gdcm::BasicOffsetTable::operator<<](#), [operator<<](#), [gdcm::Fragment::operator<<](#), [gdcm::Item::operator<<](#), [operator==\(\)](#), [gdcm::BasicOffsetTable::Read\(\)](#), [gdcm::Item::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), [gdcm::Fragment::ReadPreValue\(\)](#), [gdcm::Fragment::ReadValue\(\)](#), [SetValue\(\)](#), [SetVL\(\)](#), [gdcm::Fragment::Write\(\)](#), and [gdcm::Item::Write\(\)](#).

12.73.6.4 VRField

[VR](#) gdcm::DataElement::VRField [protected]

Referenced by [DataElement\(\)](#), [Clear\(\)](#), [GetVR\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetVR\(\)](#).

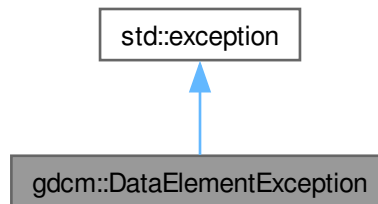
The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

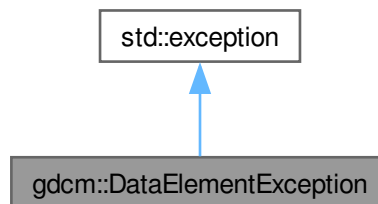
12.74 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



The documentation for this class was generated from the following file:

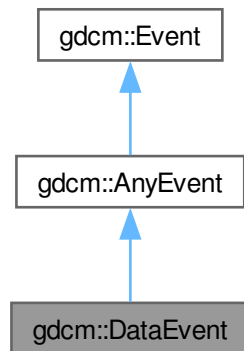
- [gdcmDataSet.h](#)

12.75 gdcm::DataEvent Class Reference

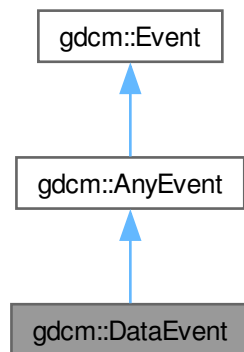
[DataEvent](#).

```
#include <gdcmDataEvent.h>
```

Inheritance diagram for gdcm::DataEvent:



Collaboration diagram for gdcm::DataEvent:



Public Types

- typedef [DataEvent](#) Self
- typedef [AnyEvent](#) Superclass

Public Member Functions

- [DataEvent](#) (const char *bytes=nullptr, size_t len=0)

- [DataEvent](#) (const [Self](#) &s)
- [~DataEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const override
- const char * [GetData](#) () const
- size_t [GetDataLength](#) () const
- const char * [GetEventName](#) () const override
- [::gdcm::Event](#) * [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetData](#) (const char *bytes, size_t len)

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

12.75.1 Detailed Description

[DataEvent](#).

12.75.2 Member Typedef Documentation

12.75.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

12.75.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

12.75.3 Constructor & Destructor Documentation

12.75.3.1 [DataEvent](#)() [1/2]

```
gdcm::DataEvent::DataEvent (  
    const char * bytes = nullptr,  
    size_t len = 0) [inline]
```


12.75.3.2 ~DataEvent()

gdcm::DataEvent::~DataEvent () [override], [default]

12.75.3.3 DataEvent() [2/2]

gdcm::DataEvent::~DataEvent (
 const [Self](#) & s) [inline]

12.75.4 Member Function Documentation

12.75.4.1 CheckEvent()

bool gdcm::DataEvent::CheckEvent (
 const [::gdcm::Event](#) * e) const [inline], [override]

12.75.4.2 GetData()

const char * gdcm::DataEvent::GetData () const [inline]

12.75.4.3 GetDataLength()

size_t gdcm::DataEvent::GetDataLength () const [inline]

12.75.4.4 GetEventName()

const char * gdcm::DataEvent::GetEventName () const [inline], [override], [virtual]

Return the StringName associated with the event.

Implements [gdcm::Event](#).

12.75.4.5 MakeObject()

[::gdcm::Event](#) * gdcm::DataEvent::MakeObject () const [inline], [override], [virtual]

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcm::Event](#).

12.75.4.6 operator=()

```
void gdcM::DataEvent::operator= (  
    const Self & ) [delete]
```

12.75.4.7 SetData()

```
void gdcM::DataEvent::SetData (  
    const char * bytes,  
    size_t len) [inline]
```

The documentation for this class was generated from the following file:

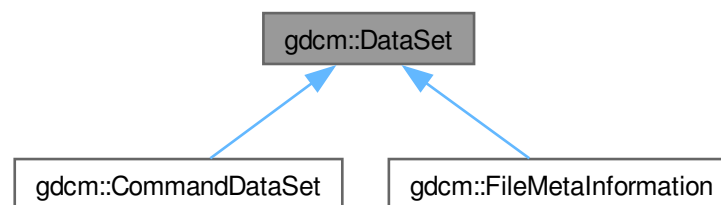
- [gdcMDataEvent.h](#)

12.76 gdcM::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements).

```
#include <gdcMDataSet.h>
```

Inheritance diagram for gdcM::DataSet:



Public Types

- typedef DataElementSet::const_iterator [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef DataElementSet::iterator [Iterator](#)
- typedef DataElementSet::size_type [SizeType](#)

Public Member Functions

- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()
- template<typename TDE>
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const
Look up if private tag 't' is present in the dataset:
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const
Return the dataelement.
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataElementSet](#) & [GetDES](#) ()
- const [DataElementSet](#) & [GetDES](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [MediaStorage GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const
- [PrivateTag GetPrivateTag](#) (const [Tag](#) &t) const
Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsEmpty](#) () const
Returns if the dataset is empty.
- const [DataElement](#) & [operator\(\)](#) (uint16_t group, uint16_t element) const
- [DataSet](#) & [operator=](#) ([DataSet](#) const &)=default
- const [DataElement](#) & [operator\[\]](#) (const [Tag](#) &t) const
- void [Print](#) (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadNested](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTags](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTagsWithLength](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTag](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTagWithLength](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags, [VL](#) &length)

- `template<typename TDE, typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `SizeType Remove (const Tag &tag)`
 Completely remove a dataelement from the dataset.
- `void Replace (const DataElement &de)`
 Replace a dataelement with another one.
- `void ReplaceEmpty (const DataElement &de)`
 Only replace a DICOM attribute when it is missing or empty.
- `SizeType Size () const`
- `template<typename TDE, typename TSwap>`
`std::ostream const & Write (std::ostream &os) const`

Protected Member Functions

- `Tag ComputeDataElement (const PrivateTag &t) const`
- `const DataElement & GetDEEnd () const`
- `void InsertDataElement (const DataElement &de)`

Friends

- class `CSAHeader`
- `std::ostream & operator<< (std::ostream &__os, const DataSet &val)`

12.76.1 Detailed Description

Class to represent a Data Set (which contains Data Elements).

A Data Set represents an instance of a real world Information [Object](#)

Note

DATA SET: Exchanged information consisting of a structured set of [Attribute](#) values directly or indirectly related to Information Objects. The value of each [Attribute](#) in a Data Set is expressed as a [Data Element](#). A collection of Data Elements ordered by increasing Data [Element Tag](#) number that is an encoding of the values of Attributes of a real world object.

Implementation note. If one do: `DataSet ds; ds.SetLength(0); ds.Read(is);` setting length to 0 actually means try to read is as if it was a root [DataSet](#). Other value are undefined (nested dataset with undefined length) or defined length (different from 0) means nested dataset with defined length.

Warning

a [DataSet](#) does not have a Transfer Syntax type, only a [File](#) does.

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [SortImage.cxx](#), [SortImage2.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [VolumeSorter.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.76.2 Member Typedef Documentation

12.76.2.1 ConstIterator

typedef DataElementSet::const_iterator [gdcm::DataSet::ConstIterator](#)

Examples

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), and [DuplicatePCDE.cxx](#).

12.76.2.2 DataElementSet

typedef std::set<[DataElement](#)> [gdcm::DataSet::DataElementSet](#)

12.76.2.3 Iterator

typedef DataElementSet::iterator [gdcm::DataSet::Iterator](#)

12.76.2.4 SizeType

typedef DataElementSet::size_type [gdcm::DataSet::SizeType](#)

12.76.3 Member Function Documentation

12.76.3.1 Begin() [1/2]

[Iterator](#) [gdcm::DataSet::Begin \(\)](#) [inline]

12.76.3.2 Begin() [2/2]

[ConstIterator](#) gdcm::DataSet::Begin () const [inline]

Examples

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), and [DuplicatePCDE.cxx](#).

12.76.3.3 Clear()

void gdcm::DataSet::Clear () [inline]

References [gdcm_assert](#).

Referenced by [gdcm::Item::Read\(\)](#).

12.76.3.4 ComputeDataElement()

[Tag](#) gdcm::DataSet::ComputeDataElement (
const [PrivateTag](#) & t) const [protected]

References [operator<<](#).

12.76.3.5 ComputeGroupLength()

template<typename TDE>
unsigned int gdcm::DataSet::ComputeGroupLength (
[Tag](#) const & tag) const [inline]

References [gdcm_assert](#), [gdcm::Tag::GetElement\(\)](#), and [gdcm::Tag::GetGroup\(\)](#).

12.76.3.6 End() [1/2]

[Iterator](#) gdcm::DataSet::End () [inline]

12.76.3.7 End() [2/2]

[ConstIterator](#) gdcm::DataSet::End () const [inline]

Examples

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpVisusChange.cxx](#), and [DuplicatePCDE.cxx](#).

12.76.3.8 FindDataElement() [1/2]

```
bool gdcmm::DataSet::FindDataElement (
    const PrivateTag & t) const
```

Look up if private tag 't' is present in the dataset:

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet\(\)](#), and [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet\(\)](#).

12.76.3.9 FindDataElement() [2/2]

```
bool gdcmm::DataSet::FindDataElement (
    const Tag & t) const [inline]
```

References [GetDataElement\(\)](#), and [GetDEEnd\(\)](#).

12.76.3.10 FindNextDataElement()

```
const DataElement & gdcmm::DataSet::FindNextDataElement (
    const Tag & t) const [inline]
```

Examples

[DuplicatePCDE.cxx](#).

References [GetDEEnd\(\)](#).

12.76.3.11 GetDataElement() [1/2]

```
const DataElement & gdcmm::DataSet::GetDataElement (
    const PrivateTag & t) const
```

Return the dataelement.

12.76.3.12 GetDataElement() [2/2]

```
const DataElement & gdcmm::DataSet::GetDataElement (
    const Tag & t) const    [inline]
```

Return the [DataElement](#) with [Tag](#) 't'

Warning

: This only search at the 'root level' of the [DataSet](#)

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#),
[DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#),
[DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#),
[DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#),
[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#),
[GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#),
[ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#),
[iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [GetDEEnd\(\)](#).

Referenced by [FindDataElement\(\)](#), [operator\(\)\(\)](#), [operator\[\]\(\)](#), [gdcmm::Attribute< Group, Element, TVR, TVM >::Set\(\)](#),
[gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Set\(\)](#),
[gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet\(\)](#), [gdcmm::Attribute< Group, Element, TVR, VM::VM1](#)
and [gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet\(\)](#).

12.76.3.13 GetDEEnd()

```
const DataElement & gdcmm::DataSet::GetDEEnd () const    [protected]
```

Referenced by [FindDataElement\(\)](#), [FindNextDataElement\(\)](#), and [GetDataElement\(\)](#).

12.76.3.14 GetDES() [1/2]

```
DataElementSet & gdcmm::DataSet::GetDES ()    [inline]
```

12.76.3.15 GetDES() [2/2]

```
const DataElementSet & gdcmm::DataSet::GetDES () const    [inline]
```

Examples

[ReadAndDumpDICOMDIR.cxx](#).

12.76.3.16 GetLength()

```
template<typename TDE>
```

```
VL gdcm::DataSet::GetLength () const [inline]
```

References [gdcm_assert](#).

Referenced by [gdcm::FileMetaInformation::GetFullLength\(\)](#).

12.76.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage () const
```

12.76.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (
    const Tag & t) const
```

Return the private creator of the private tag 't': or an empty string when not found

Examples

[DuplicatePCDE.cxx](#).

12.76.3.19 GetPrivateTag()

```
PrivateTag gdcm::DataSet::GetPrivateTag (
    const Tag & t) const
```

Return the private tag of the private tag 't', private creator will be set to empty if not found.

12.76.3.20 Insert()

```
void gdcm::DataSet::Insert (
    const DataElement & de) [inline]
```

Insert a [DataElement](#) in the [DataSet](#).

Warning

: [Tag](#) need to be $\geq 0x8$ to be considered valid data element

Examples

[CreateJPIPDataSet.cxx](#), [DumpSiemensBase64.cxx](#), [DuplicatePCDE.cxx](#), [Extracting__All__Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DataElement::GetTag\(\)](#), and [InsertDataElement\(\)](#).

12.76.3.21 InsertDataElement()

```
void gdcM::DataSet::InsertDataElement (
    const DataElement & de) [inline], [protected]
```

References [gdcM_assert](#), [gdcMWarningMacro](#), [gdcM::Value::GetLength\(\)](#), [gdcM::DataElement::GetValue\(\)](#), [gdcM::DataElement::GetVL\(\)](#), and [gdcM::DataElement::IsEmpty\(\)](#).

Referenced by [gdcM::CommandDataSet::Insert\(\)](#), [Insert\(\)](#), and [gdcM::FileMetaInformation::Insert\(\)](#).

12.76.3.22 IsEmpty()

```
bool gdcM::DataSet::IsEmpty () const [inline]
```

Returns if the dataset is empty.

Referenced by [gdcM::Item::Read\(\)](#).

12.76.3.23 operator()()

```
const DataElement & gdcM::DataSet::operator() (
    uint16_t group,
    uint16_t element) const [inline]
```

References [GetDataElement\(\)](#).

12.76.3.24 operator=()

```
DataSet & gdcM::DataSet::operator= (
    DataSet const & ) [default]
```

12.76.3.25 operator[]()

```
const DataElement & gdcM::DataSet::operator[] (
    const Tag & t) const [inline]
```

References [GetDataElement\(\)](#).

12.76.3.26 Print()

```
void gdcM::DataSet::Print (
    std::ostream & os,
    std::string const & indent = "") const [inline]
```

Referenced by [gdcM::CommandDataSet::operator<<](#), [operator<<](#), [gdcM::FileMetaInformation::operator<<](#), and [gdcM::Item::operator<<](#).

12.76.3.27 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::Read (
    std::istream & is)
```

Examples

[DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

12.76.3.28 ReadNested()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadNested (
    std::istream & is)
```

12.76.3.29 ReadSelectedPrivateTags()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadSelectedPrivateTags (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    bool readvalues = true)
```

12.76.3.30 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadSelectedPrivateTagsWithLength (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    VL & length,
    bool readvalues = true)
```

12.76.3.31 ReadSelectedTags()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadSelectedTags (
    std::istream & is,
    const std::set< Tag > & tags,
    bool readvalues = true)
```

12.76.3.32 ReadSelectedTagsWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadSelectedTagsWithLength (
    std::istream & is,
    const std::set< Tag > & tags,
    VL & length,
    bool readvalues = true)
```

12.76.3.33 ReadUpToTag()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadUpToTag (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags)
```

12.76.3.34 ReadUpToTagWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadUpToTagWithLength (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags,
    VL & length)
```

12.76.3.35 ReadWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcmm::DataSet::ReadWithLength (
    std::istream & is,
    VL & length)
```

12.76.3.36 Remove()

```
SizeType gdcmm::DataSet::Remove (
    const Tag & tag) [inline]
```

Completely remove a dataelement from the dataset.

Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [gdcmm_assert](#).

Referenced by [gdcmm::CommandDataSet::Replace\(\)](#), and [gdcmm::FileMetaInformation::Replace\(\)](#).

12.76.3.37 Replace()

```
void gdcm::DataSet::Replace (  
    const DataElement & de) [inline]
```

Replace a dataelement with another one.

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#),
[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#),
[HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and
[rle2img.cxx](#).

References [gdcmAssertAlwaysMacro](#).

12.76.3.38 ReplaceEmpty()

```
void gdcm::DataSet::ReplaceEmpty (  
    const DataElement & de) [inline]
```

Only replace a DICOM attribute when it is missing or empty.

Examples

[rle2img.cxx](#).

References [gdcmAssertAlwaysMacro](#).

12.76.3.39 Size()

```
SizeType gdcm::DataSet::Size () const [inline]
```

Examples

[DumpGEMSMovieGroup.cxx](#).

Referenced by [gdcm::SequenceOfItems::Read\(\)](#).

12.76.3.40 Write()

```
template<typename TDE, typename TSwap>  
std::ostream const & gdcm::DataSet::Write (  
    std::ostream & os) const
```

12.76.4 Friends And Related Symbol Documentation

12.76.4.1 CSAHeader

friend class CSAHeader [friend]

References [CSAHeader](#).

Referenced by [CSAHeader](#).

12.76.4.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const DataSet & val) [friend]
```

References [operator<<](#), and [Print\(\)](#).

Referenced by [ComputeDataElement\(\)](#), and [operator<<](#).

The documentation for this class was generated from the following file:

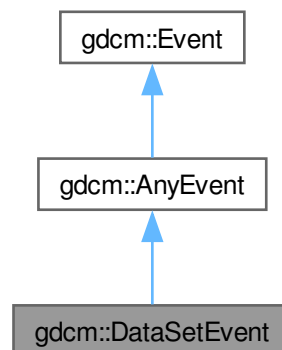
- [gdcmDataSet.h](#)

12.77 gdcm::DataSetEvent Class Reference

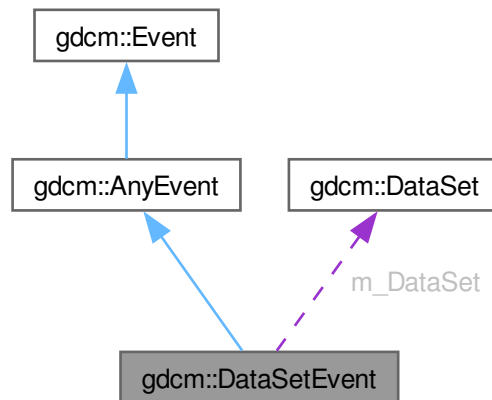
[DataSetEvent](#).

```
#include <gdcmDataSetEvent.h>
```

Inheritance diagram for gdcm::DataSetEvent:



Collaboration diagram for gdcm::DataSetEvent:



Public Types

- typedef [DataSetEvent](#) [Self](#)
- typedef [AnyEvent](#) [Superclass](#)

Public Member Functions

- [DataSetEvent](#) (const [Self](#) &s)
- [DataSetEvent](#) ([DataSet](#) const *ds=nullptr)
- [~DataSetEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const override
- [DataSet](#) const & [GetDataSet](#) () const
- const char * [GetEventName](#) () const override
- [::gdcm::Event](#) * [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

Public Attributes

- const [DataSet](#) * [m_DataSet](#)

12.77.1 Detailed Description

[DataSetEvent](#).

Special type of event triggered during the [DataSet](#) store/move process

12.77.2 Member Typedef Documentation

12.77.2.1 Self

```
typedef DataSetEvent gdcm::DataSetEvent::Self
```

12.77.2.2 Superclass

```
typedef AnyEvent gdcm::DataSetEvent::Superclass
```

12.77.3 Constructor & Destructor Documentation

12.77.3.1 [DataSetEvent\(\)](#) [1/2]

```
gdcm::DataSetEvent::DataSetEvent (  
    DataSet const * ds = nullptr) [inline]
```

References [m_DataSet](#).

12.77.3.2 [~DataSetEvent\(\)](#)

```
gdcm::DataSetEvent::~~DataSetEvent () [override], [default]
```

12.77.3.3 [DataSetEvent\(\)](#) [2/2]

```
gdcm::DataSetEvent::DataSetEvent (  
    const Self & s) [inline]
```


12.77.4 Member Function Documentation

12.77.4.1 CheckEvent()

```
bool gdcm::DataSetEvent::CheckEvent (
    const ::gdcm::Event * e) const    [inline], [override]
```

12.77.4.2 GetDataSet()

```
DataSet const & gdcm::DataSetEvent::GetDataSet () const    [inline]
```

References [m_DataSet](#).

12.77.4.3 GetEventName()

```
const char * gdcm::DataSetEvent::GetEventName () const    [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

12.77.4.4 MakeObject()

```
::gdcm::Event * gdcm::DataSetEvent::MakeObject () const    [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcm::Event](#).

12.77.4.5 operator=()

```
void gdcm::DataSetEvent::operator= (
    const Self & )    [delete]
```

12.77.5 Member Data Documentation

12.77.5.1 m_DataSet

```
const DataSet* gdcm::DataSetEvent::m_DataSet
```

Referenced by [DataSetEvent\(\)](#), and [GetDataSet\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmDataSetEvent.h](#)

12.78 gdcm::DataSetHelper Class Reference

[DataSetHelper](#) (internal class, not intended for user level).

```
#include <gdcmDataSetHelper.h>
```

Static Public Member Functions

- static [VR](#) [ComputeVR](#) ([File](#) const &file, [DataSet](#) const &ds, const [Tag](#) &tag)

12.78.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level).

Examples

[SimplePrint.cs](#).

12.78.2 Member Function Documentation

12.78.2.1 ComputeVR()

```
VR gdcm::DataSetHelper::ComputeVR (  
    File const & file,  
    DataSet const & ds,  
    const Tag & tag) [static]
```

ds -> current dataset, which is not the same as the root dataset return [VR::INVALID](#) in case of error

Examples

[SimplePrint.cs](#).

The documentation for this class was generated from the following file:

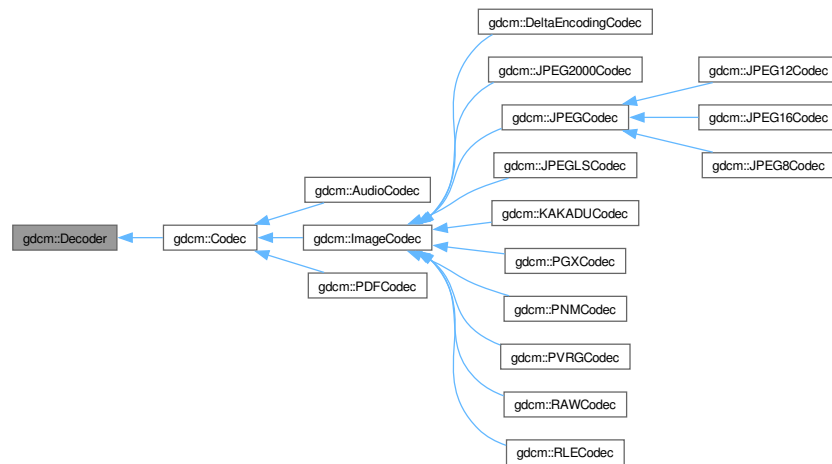
- [gdcmDataSetHelper.h](#)

12.79 gdcm::Decoder Class Reference

[Decoder](#).

```
#include <gdcmDecoder.h>
```

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual [~Decoder](#) ()=default
- virtual bool [CanDecode](#) ([TransferSyntax](#) const &) const =0
Return whether this decoder support this transfer syntax (can decode it).
- virtual bool [Decode](#) ([DataElement](#) const &, [DataElement](#) &)
Decode.

Protected Member Functions

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

12.79.1 Detailed Description

[Decoder](#).

12.79.2 Constructor & Destructor Documentation

12.79.2.1 ~Decoder()

virtual gdcm::Decoder::~~Decoder () [virtual], [default]

12.79.3 Member Function Documentation

12.79.3.1 CanDecode()

```
virtual bool gdcm::Decoder::CanDecode (
    TransferSyntax const & ) const    [pure virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implemented in [gdcm::AudioCodec](#), [gdcm::ImageCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PDFCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

12.79.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
    DataElement const & ,
    DataElement & )    [inline], [virtual]
```

Decode.

Reimplemented in [gdcm::AudioCodec](#), [gdcm::DeltaEncodingCodec](#), [gdcm::ImageCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::KAKADUCodec](#), [gdcm::PDFCodec](#), [gdcm::PVRGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

12.79.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
    std::istream & ,
    std::ostream & )    [inline], [protected], [virtual]
```

Reimplemented in [gdcm::ImageCodec](#), [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG8Codec](#), [gdcm::JPEGCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

The documentation for this class was generated from the following file:

- [gdcmDecoder.h](#)

12.80 gdcm::DefinedTerms Class Reference

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.

```
#include <gdcmDefinedTerms.h>
```

Public Member Functions

- [DefinedTerms](#) ()=default

12.80.1 Detailed Description

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.

12.80.2 Constructor & Destructor Documentation

12.80.2.1 DefinedTerms()

gdcm::DefinedTerms::DefinedTerms () [default]

The documentation for this class was generated from the following file:

- [gdcmDefinedTerms.h](#)

12.81 gdcm::Defs Class Reference

FIXME I do not like the name '[Defs](#)'.

```
#include <gdcmDefs.h>
```

Public Member Functions

- [Defs](#) ()
- [Defs](#) (const [Defs](#) &val)=delete
- [~Defs](#) ()
- const [IOD](#) & [GetIODFromFile](#) (const [File](#) &file) const
- [IODs](#) & [GetIODs](#) ()
- const [IODs](#) & [GetIODs](#) () const
- [Macros](#) & [GetMacros](#) ()
- const [Macros](#) & [GetMacros](#) () const
- [Modules](#) & [GetModules](#) ()
- const [Modules](#) & [GetModules](#) () const
- [Type](#) [GetTypeFromTag](#) (const [File](#) &file, const [Tag](#) &tag) const
- bool [IsEmpty](#) () const
- [Defs](#) & [operator=](#) (const [Defs](#) &val)=delete
- bool [Verify](#) (const [DataSet](#) &ds) const
- bool [Verify](#) (const [File](#) &file) const

Static Public Member Functions

- static const char * [GetIODNameFromMediaStorage](#) ([MediaStorage](#) const &ms)

Protected Member Functions

- void [LoadDefaults](#) ()
- void [LoadFromFile](#) (const char *filename)

Friends

- class [Global](#)

12.81.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

Note

bla

Examples

[GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

12.81.2 Constructor & Destructor Documentation

12.81.2.1 [Defs\(\)](#) [1/2]

gdcm::Defs::Defs ()

Referenced by [Defs\(\)](#), and [operator=\(\)](#).

12.81.2.2 [~Defs\(\)](#)

gdcm::Defs::~Defs ()

12.81.2.3 [Defs\(\)](#) [2/2]

gdcm::Defs::Defs (
 const Defs & val) [delete]

References [Defs\(\)](#).

12.81.3 Member Function Documentation

12.81.3.1 GetIODFromFile()

```
const IOD & gdcm::Defs::GetIODFromFile (  
    const File & file) const
```

12.81.3.2 GetIODNameFromMediaStorage()

```
const char * gdcm::Defs::GetIODNameFromMediaStorage (  
    MediaStorage const & ms) [static]
```

Examples

[GenerateStandardSOPClasses.cxx](#).

12.81.3.3 GetIODs() [1/2]

```
IODs & gdcm::Defs::GetIODs () [inline]
```

12.81.3.4 GetIODs() [2/2]

```
const IODs & gdcm::Defs::GetIODs () const [inline]
```

Examples

[TraverseModules.cxx](#).

12.81.3.5 GetMacros() [1/2]

```
Macros & gdcm::Defs::GetMacros () [inline]
```

12.81.3.6 GetMacros() [2/2]

```
const Macros & gdcm::Defs::GetMacros () const [inline]
```

Users should not directly use [Macro](#). [Macro](#) are simply a way for DICOM WG to re-use Tables. [Macros](#) are conveniently wrapped within [Modules](#). See [gdcm::Module](#) API directly

Examples

[TraverseModules.cxx](#).

12.81.3.7 GetModules() [1/2]

[Modules](#) & gdcm::Defs::GetModules () [inline]

12.81.3.8 GetModules() [2/2]

const [Modules](#) & gdcm::Defs::GetModules () const [inline]

Examples

[TraverseModules.cxx](#).

Referenced by [IsEmpty\(\)](#).

12.81.3.9 GetTypeFromTag()

[Type](#) gdcm::Defs::GetTypeFromTag (
 const [File](#) & file,
 const [Tag](#) & tag) const

12.81.3.10 IsEmpty()

bool gdcm::Defs::IsEmpty () const [inline]

References [GetModules\(\)](#).

12.81.3.11 LoadDefaults()

void gdcm::Defs::LoadDefaults () [protected]

12.81.3.12 LoadFromFile()

void gdcm::Defs::LoadFromFile (
 const char * filename) [protected]

12.81.3.13 operator=()

[Defs](#) & gdcm::Defs::operator= (
 const [Defs](#) & val) [delete]

References [Defs\(\)](#).

12.81.3.14 Verify() [1/2]

```
bool gdcm::Defs::Verify (  
    const DataSet & ds) const
```

12.81.3.15 Verify() [2/2]

```
bool gdcm::Defs::Verify (  
    const File & file) const
```

12.81.4 Friends And Related Symbol Documentation

12.81.4.1 Global

```
friend class Global [friend]
```

References [Global](#).

Referenced by [Global](#).

The documentation for this class was generated from the following file:

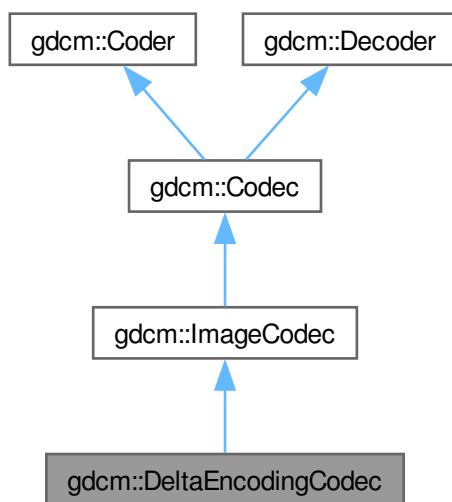
- [gdcmDefs.h](#)

12.82 gdcm::DeltaEncodingCodec Class Reference

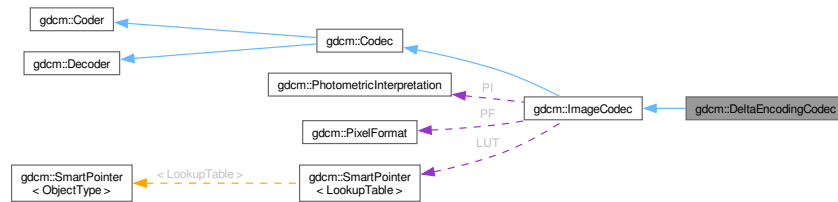
[DeltaEncodingCodec](#) compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for gdcm::DeltaEncodingCodec:



Collaboration diagram for `gdcm::DeltaEncodingCodec`:



Public Member Functions

- [DeltaEncodingCodec](#) ()
 - [~DeltaEncodingCodec](#) ()
 - bool [CanDecode](#) ([TransferSyntax](#) const &ts)
 - bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
- Decode.

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- virtual [ImageCodec](#) * [Clone](#) () const =0
- bool [Decode](#) ([DataElement](#) const &is_, [DataElement](#) &os) override
- Decode.
- const unsigned int * [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_) Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [Decode](#) (std::istream &is, std::ostream &os)

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.82.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

12.82.2 Constructor & Destructor Documentation

12.82.2.1 [DeltaEncodingCodec\(\)](#)

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ()
```

12.82.2.2 [~DeltaEncodingCodec\(\)](#)

```
gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ()
```

12.82.3 Member Function Documentation

12.82.3.1 [CanDecode\(\)](#)

```
bool gdcm::DeltaEncodingCodec::CanDecode (
    TransferSyntax const & ts)
```

12.82.3.2 [Decode\(\)](#) [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

12.82.3.3 Decode() [2/2]

```
bool gdcm::DeltaEncodingCodec::Decode (  
    std::istream & is,  
    std::ostream & os) [protected]
```

The documentation for this class was generated from the following file:

- [gdcmDeltaEncodingCodec.h](#)

12.83 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- [DICOMDIR](#) ()=default
- [DICOMDIR](#) ([FileSet](#) fs)

12.83.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

12.83.2 Constructor & Destructor Documentation

12.83.2.1 DICOMDIR() [1/2]

```
gdcm::DICOMDIR::DICOMDIR () [default]
```

12.83.2.2 DICOMDIR() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (  
    FileSet fs) [inline]
```

The documentation for this class was generated from the following file:

- [gdcmDICOMDIR.h](#)

12.84 gdcm::DICOMDIRGenerator Class Reference

[DICOMDIRGenerator](#) class.

```
#include <gdcmDICOMDIRGenerator.h>
```

Public Types

- typedef [Directory::FileNamesType](#) FileNamesType
- typedef [Directory::FilenameType](#) FilenameType

Public Member Functions

- [DICOMDIRGenerator](#) ()
- [~DICOMDIRGenerator](#) ()
- bool [Generate](#) ()
Main function to generate the [DICOMDIR](#).
- [File](#) & [GetFile](#) ()
- void [SetDescriptor](#) (const char *d)
- void [SetFile](#) (const [File](#) &f)
Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.
- void [SetFileNames](#) ([FileNamesType](#) const &fns)
Set the list of filenames from which the [DICOMDIR](#) should be generated from.
- void [SetRootDirectory](#) ([FilenameType](#) const &root)
Set the root directory from which the filenames should be considered.

Protected Member Functions

- bool [AddImageDirectoryRecord](#) ()
- bool [AddPatientDirectoryRecord](#) ()
- bool [AddSeriesDirectoryRecord](#) ()
- bool [AddStudyDirectoryRecord](#) ()
- [Scanner](#) & [GetScanner](#) ()

12.84.1 Detailed Description

[DICOMDIRGenerator](#) class.

This is a STD-GEN-CD [DICOMDIR](#) generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles

Note

PS 3.11 - 2008 / D.3.2 Physical Medium And Medium Format The STD-GEN-CD and STD-GEN-SEC-CD application profiles require the 120 mm CD-R physical medium with the ISO/IEC 9660 Media Format, as defined in PS3.12. See also PS 3.12 - 2008 / Annex F 120mm CD-R Medium (Normative) and PS 3.10 - 2008 / 8 DICOM [File](#) Service / 8.1 FILE-SET

Warning

: PS 3.11 - 2008 / D.3.1 SOP Classes and Transfer Syntaxes Composite [Image](#) & Stand-alone Storage are required to be stored as Explicit [VR](#) Little Endian Uncompressed (1.2.840.10008.1.2.1). When a DICOM file is found using another Transfer Syntax the generator will simply stops.

- Input files should be Explicit [VR](#) Little Endian
- filenames should be valid [VR::CS](#) value (16 bytes, upper case ...)

[Bug](#) : There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the [Scanner](#) does not allow us See PS 3.11 / [Table](#) D.3-2 STD-GEN Additional [DICOMDIR](#) Keys

Examples

[GenerateDICOMDIR.cs](#).

12.84.2 Member Typedef Documentation

12.84.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

12.84.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

12.84.3 Constructor & Destructor Documentation

12.84.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ()
```

12.84.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ()
```

12.84.4 Member Function Documentation

12.84.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord () [protected]
```

12.84.4.2 AddPatientDirectoryRecord()

bool gdcmm::DICOMDIRGenerator::AddPatientDirectoryRecord () [protected]

12.84.4.3 AddSeriesDirectoryRecord()

bool gdcmm::DICOMDIRGenerator::AddSeriesDirectoryRecord () [protected]

12.84.4.4 AddStudyDirectoryRecord()

bool gdcmm::DICOMDIRGenerator::AddStudyDirectoryRecord () [protected]

12.84.4.5 Generate()

bool gdcmm::DICOMDIRGenerator::Generate ()

Main function to generate the [DICOMDIR](#).

Examples

[GenerateDICOMDIR.cs](#).

12.84.4.6 GetFile()

[File](#) & gdcmm::DICOMDIRGenerator::GetFile ()

Examples

[GenerateDICOMDIR.cs](#).

12.84.4.7 GetScanner()

[Scanner](#) & gdcmm::DICOMDIRGenerator::GetScanner () [protected]

12.84.4.8 SetDescriptor()

void gdcmm::DICOMDIRGenerator::SetDescriptor (
 const char * d)

Set the [File](#) Set ID.

Warning

 this need to be a valid [VR::CS](#) value

Examples

[GenerateDICOMDIR.cs](#).

12.84.4.9 SetFile()

```
void gdcm::DICOMDIRGenerator::SetFile (  
    const File & f)
```

Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.

12.84.4.10 SetFilenames()

```
void gdcm::DICOMDIRGenerator::SetFilenames (  
    FilenamesType const & fns)
```

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

Examples

[GenerateDICOMDIR.cs](#).

12.84.4.11 SetRootDirectory()

```
void gdcm::DICOMDIRGenerator::SetRootDirectory (  
    FilenameType const & root)
```

Set the root directory from which the filenames should be considered.

The documentation for this class was generated from the following file:

- [gdcmDICOMDIRGenerator.h](#)

12.85 gdcm::Dict Class Reference

Class to represent a map of [DictEntry](#).

```
#include <gdcmDict.h>
```

Public Types

- typedef MapDictEntry::const_iterator [ConstIterator](#)
- typedef MapDictEntry::iterator [Iterator](#)
- typedef std::map< [Tag](#), [DictEntry](#) > [MapDictEntry](#)

Public Member Functions

- [Dict](#) ()
- [Dict](#) (const [Dict](#) &_val)=delete
- void [AddDictEntry](#) (const [Tag](#) &tag, const [DictEntry](#) &de)
- [ConstIterator Begin](#) () const
- [ConstIterator End](#) () const
- const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag) const
- const [DictEntry](#) & [GetDictEntryByKeyword](#) (const char *keyword, [Tag](#) &tag) const
- const [DictEntry](#) & [GetDictEntryByName](#) (const char *name, [Tag](#) &tag) const
- const char * [GetKeywordFromTag](#) ([Tag](#) const &tag) const

Function to return the Keyword from a [Tag](#).

- bool [IsEmpty](#) () const
- [Dict](#) & [operator=](#) (const [Dict](#) &_val)=delete

Protected Member Functions

- void [LoadDefault](#) ()

Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &__os, const [Dict](#) &_val)

12.85.1 Detailed Description

Class to represent a map of [DictEntry](#).

Note

bla TODO FIXME: For [Element](#) == 0x0 need to return Name = Group Length ValueRepresentation = UL ValueMultiplicity = 1

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

12.85.2 Member Typedef Documentation

12.85.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcmm::Dict::ConstIterator
```

12.85.2.2 Iterator

typedef MapDictEntry::iterator [gdcmm::Dict::Iterator](#)

12.85.2.3 MapDictEntry

typedef std::map<[Tag](#), [DictEntry](#)> [gdcmm::Dict::MapDictEntry](#)

12.85.3 Constructor & Destructor Documentation

12.85.3.1 Dict() [1/2]

[gdcmm::Dict::Dict](#) () [inline]

References [gdcmm_assert](#).

Referenced by [Dict\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.85.3.2 Dict() [2/2]

[gdcmm::Dict::Dict](#) (
 const [Dict](#) & _val) [delete]

References [Dict\(\)](#), and [operator<<](#).

12.85.4 Member Function Documentation

12.85.4.1 AddDictEntry()

void [gdcmm::Dict::AddDictEntry](#) (
 const [Tag](#) & tag,
 const [DictEntry](#) & de) [inline]

References [gdcmm_assert](#).

12.85.4.2 Begin()

[ConstIterator](#) [gdcmm::Dict::Begin](#) () const [inline]

Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

12.85.4.3 End()

`ConstIterator` `gdcm::Dict::End () const` [inline]

Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

12.85.4.4 GetDictEntry()

`const DictEntry & gdcm::Dict::GetDictEntry (`
`const Tag & tag) const` [inline]

Examples

[GenFakeIdentifyFile.cxx](#), and [PublicDict.cxx](#).

References [gdcm_assert](#).

12.85.4.5 GetDictEntryByKeyword()

`const DictEntry & gdcm::Dict::GetDictEntryByKeyword (`
`const char * keyword,`
`Tag & tag) const` [inline]

Lookup [DictEntry](#) by keyword. Even if DICOM standard defines keyword as being unique. The lookup table is built on [Tag](#). Therefore looking up a [DictEntry](#) by Keyword is more inefficient than looking up by [Tag](#).

References [gdcm_assert](#).

12.85.4.6 GetDictEntryByName()

`const DictEntry & gdcm::Dict::GetDictEntryByName (`
`const char * name,`
`Tag & tag) const` [inline]

Inefficient way of looking up tag by name. Technically DICOM does not guarantee uniqueness (and [Curve](#) / [Overlay](#) are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

Examples

[ReadAndPrintAttributes.cxx](#).

References [gdcm_assert](#).

12.85.4.7 GetKeywordFromTag()

```
const char * gdcmm::Dict::GetKeywordFromTag (  
    Tag const & tag) const    [inline]
```

Function to return the Keyword from a [Tag](#).

References [gdcmm_assert](#).

12.85.4.8 IsEmpty()

```
bool gdcmm::Dict::IsEmpty () const    [inline]
```

12.85.4.9 LoadDefault()

```
void gdcmm::Dict::LoadDefault ()    [protected]
```

12.85.4.10 operator=()

```
Dict & gdcmm::Dict::operator= (  
    const Dict & _val)    [delete]
```

References [Dict\(\)](#).

12.85.5 Friends And Related Symbol Documentation

12.85.5.1 Dicts

```
friend class Dicts    [friend]
```

References [Dicts](#).

Referenced by [Dicts](#).

12.85.5.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Dict & _val)    [friend]
```

References [Dict\(\)](#).

Referenced by [Dict\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmDict.h](#)

12.86 gdcm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmDictConverter.h>
```

Public Types

- enum [OutputTypes](#) {
 [DICT_DEFAULT](#) = 0 ,
 [DICT_DEBUG](#) ,
 [DICT_XML](#) }

Public Member Functions

- [DictConverter](#) ()
- [~DictConverter](#) ()
- void [Convert](#) ()
- const std::string & [GetDictName](#) () const
- const std::string & [GetInputFilename](#) () const
- const std::string & [GetOutputFilename](#) () const
- int [GetOutputType](#) () const
- void [SetDictName](#) (const char *name)
- void [SetInputFileName](#) (const char *filename)
- void [SetOutputFileName](#) (const char *filename)
- void [SetOutputType](#) (int type)

Static Public Member Functions

- static bool [Readuint16](#) (const char *raw, uint16_t &ov)
- static bool [ReadVM](#) (const char *raw, [VM::VMType](#) &type)
- static bool [ReadVR](#) (const char *raw, [VR::VRType](#) &type)

Protected Member Functions

- void [AddGroupLength](#) ()
- bool [ConvertToCXX](#) (const char *raw, std::string &cxx)
- bool [ConvertToXML](#) (const char *raw, std::string &cxx)
- void [WriteFooter](#) ()
- void [WriteHeader](#) ()

12.86.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embed dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

Note

12.86.2 Member Enumeration Documentation

12.86.2.1 OutputTypes

enum [gdcmm::DictConverter::OutputTypes](#)

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

12.86.3 Constructor & Destructor Documentation

12.86.3.1 DictConverter()

[gdcmm::DictConverter::DictConverter](#) ()

12.86.3.2 ~DictConverter()

[gdcmm::DictConverter::~~DictConverter](#) ()

12.86.4 Member Function Documentation

12.86.4.1 AddGroupLength()

[void gdcmm::DictConverter::AddGroupLength](#) () [protected]

12.86.4.2 Convert()

```
void gdcm::DictConverter::Convert ()
```

12.86.4.3 ConvertToCXX()

```
bool gdcm::DictConverter::ConvertToCXX (  
    const char * raw,  
    std::string & cxx) [protected]
```

12.86.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (  
    const char * raw,  
    std::string & cxx) [protected]
```

12.86.4.5 GetDictName()

```
const std::string & gdcm::DictConverter::GetDictName () const
```

12.86.4.6 GetInputFilename()

```
const std::string & gdcm::DictConverter::GetInputFilename () const
```

12.86.4.7 GetOutputFilename()

```
const std::string & gdcm::DictConverter::GetOutputFilename () const
```

12.86.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType () const [inline]
```

12.86.4.9 Readuint16()

```
bool gdcm::DictConverter::Readuint16 (  
    const char * raw,  
    uint16_t & ov) [static]
```


12.86.4.10 ReadVM()

```
bool gdcm::DictConverter::ReadVM (
    const char * raw,
    VM::VMType & type) [static]
```

12.86.4.11 ReadVR()

```
bool gdcm::DictConverter::ReadVR (
    const char * raw,
    VR::VRType & type) [static]
```

12.86.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
    const char * name)
```

12.86.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
    const char * filename)
```

12.86.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
    const char * filename)
```

12.86.4.15 SetOutputType()

```
void gdcm::DictConverter::SetOutputType (
    int type) [inline]
```

12.86.4.16 WriteFooter()

```
void gdcm::DictConverter::WriteFooter () [protected]
```

12.86.4.17 WriteHeader()

```
void gdcm::DictConverter::WriteHeader () [protected]
```

The documentation for this class was generated from the following file:

- [gdcmDictConverter.h](#)

12.87 gdcm::DictEntry Class Reference

Class to represent an Entry in the [Dict](#).

```
#include <gdcmDictEntry.h>
```

Public Member Functions

- [DictEntry](#) (const char *name="", const char *keyword="", [VR](#) const &vr=[VR::INVALID](#), [VM](#) const &vm=[VM::VM0](#), bool ret=false)
- const char * [GetKeyword](#) () const
same as GetName but without spaces...
- const char * [GetName](#) () const
Set/Get Name.
- bool [GetRetired](#) () const
Set/Get Retired flag.
- const [VM](#) & [GetVM](#) () const
Set/Get [VM](#).
- const [VR](#) & [GetVR](#) () const
Set/Get [VR](#).
- bool [IsUnique](#) () const
- void [SetElementXX](#) (bool v)
Set whether element is shared in multiple elements (Source [Image](#) IDs typically).
- void [SetGroupXX](#) (bool v)
Set whether element is shared in multiple groups (Curve/Overlay typically).
- void [SetKeyword](#) (const char *keyword)
- void [SetName](#) (const char *name)
- void [SetRetired](#) (bool retired)
- void [SetVM](#) ([VM](#) const &vm)
- void [SetVR](#) (const [VR](#) &vr)

Friends

- class [Dict](#)
- std::ostream & [operator<<](#) (std::ostream &_os, const [DictEntry](#) &_val)

12.87.1 Detailed Description

Class to represent an Entry in the [Dict](#).

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdcm::Tag](#) to the needed information

Note

bla TODO FIXME: Need a PublicDictEntry...indeed [DictEntry](#) has a notion of retired which does not exist in PrivateDictEntry...

See also

[gdcm::Dict](#)

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

12.87.2 Constructor & Destructor Documentation

12.87.2.1 DictEntry()

```
gdcm::DictEntry::DictEntry (  
    const char * name = "",  
    const char * keyword = "",  
    VR const & vr = VR::INVALID,  
    VM const & vm = VM::VM0,  
    bool ret = false) [inline]
```

References [gdcm::VR::INVALID](#), and [gdcm::VM::VM0](#).

Referenced by [operator<<](#).

12.87.3 Member Function Documentation

12.87.3.1 GetKeyword()

```
const char * gdcm::DictEntry::GetKeyword () const [inline]
```

same as GetName but without spaces...

12.87.3.2 GetName()

```
const char * gdcm::DictEntry::GetName () const [inline]
```

Set/Get Name.

Referenced by [gdcm::PrivateDict::PrintXML\(\)](#).

12.87.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired () const [inline]
```

Set/Get Retired flag.

Examples

[GenAllVR.cxx](#).

12.87.3.4 GetVM()

```
const VM & gdcm::DictEntry::GetVM () const [inline]
```

Set/Get VM.

Referenced by [gdcm::PrivateDict::AddDictEntry\(\)](#), and [gdcm::PrivateDict::PrintXML\(\)](#).

12.87.3.5 GetVR()

```
const VR & gdcM::DictEntry::GetVR () const [inline]
```

Set/Get [VR](#).

Examples

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

Referenced by [gdcM::PrivateDict::AddDictEntry\(\)](#), and [gdcM::PrivateDict::PrintXML\(\)](#).

12.87.3.6 IsUnique()

```
bool gdcM::DictEntry::IsUnique () const [inline]
```

Return whether the name of the [DataElement](#) can be considered to be unique. As of 2008 all elements name were unique (except the explicitly 'XX' ones)

12.87.3.7 SetElementXX()

```
void gdcM::DictEntry::SetElementXX (  
    bool v) [inline]
```

Set whether element is shared in multiple elements (Source [Image](#) IDs typically).

12.87.3.8 SetGroupXX()

```
void gdcM::DictEntry::SetGroupXX (  
    bool v) [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically).

12.87.3.9 SetKeyword()

```
void gdcM::DictEntry::SetKeyword (  
    const char * keyword) [inline]
```

12.87.3.10 SetName()

```
void gdcM::DictEntry::SetName (  
    const char * name) [inline]
```

12.87.3.11 SetRetired()

```
void gdcmm::DictEntry::SetRetired (  
    bool retired) [inline]
```

12.87.3.12 SetVM()

```
void gdcmm::DictEntry::SetVM (  
    VM const & vm) [inline]
```

Referenced by [gdcmm::PrivateDict::AddDictEntry\(\)](#).

12.87.3.13 SetVR()

```
void gdcmm::DictEntry::SetVR (  
    const VR & vr) [inline]
```

Referenced by [gdcmm::PrivateDict::AddDictEntry\(\)](#).

12.87.4 Friends And Related Symbol Documentation

12.87.4.1 Dict

```
friend class Dict [friend]
```

References [Dict](#).

Referenced by [Dict](#).

12.87.4.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const DictEntry & __val) [friend]
```

References [DictEntry\(\)](#).

The documentation for this class was generated from the following file:

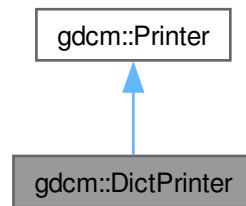
- [gdcmmDictEntry.h](#)

12.88 gdcmm::DictPrinter Class Reference

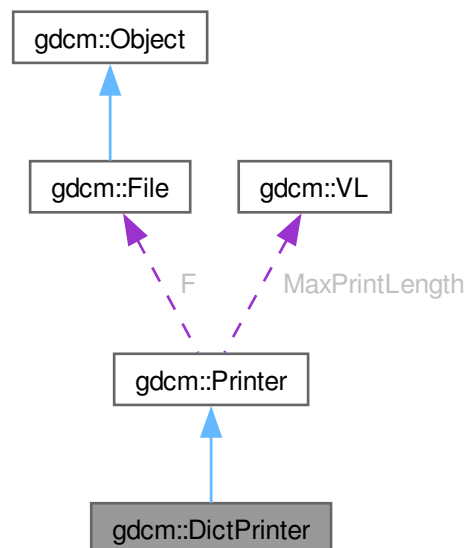
[DictPrinter](#) class.

```
#include <gdcmmDictPrinter.h>
```

Inheritance diagram for gdcmm::DictPrinter:



Collaboration diagram for gdcmm::DictPrinter:



Public Member Functions

- [DictPrinter](#) ()
- [~DictPrinter](#) ()=default
- void [Print](#) (std::ostream &os)

Public Member Functions inherited from [gdcm::Printer](#)

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles](#) [GetPrintStyle](#) () const
Get PrintStyle value.
- void [Print](#) (std::ostream &os)
Print.
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &s="")
Print an individual dataset.
- void [SetColor](#) (bool c)
Set color mode or not.
- void [SetFile](#) ([File](#) const &f)
Set file.
- void [SetStyle](#) ([PrintStyles](#) ps)
Set PrintStyle value.

Protected Member Functions

- void [PrintDataElement2](#) (std::ostream &os, const [DataSet](#) &ds, const [DataElement](#) &ide)
- void [PrintDataSet2](#) (std::ostream &os, const [DataSet](#) &ds)

Protected Member Functions inherited from [gdcm::Printer](#)

- [VR](#) [PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) *sqi, std::ostream &os, std::string const &indent)

Additional Inherited Members

Public Types inherited from [gdcm::Printer](#)

- enum [PrintStyles](#) {
 [VERBOSE_STYLE](#) = 0 ,
 [CONDENSED_STYLE](#) ,
 [XML](#) ,
 [CXX](#) }

Protected Attributes inherited from [gdcM::Printer](#)

- `const File * F`
- `VL MaxPrintLength`
- `PrintStyles PrintStyle`

12.88.1 Detailed Description

[DictPrinter](#) class.

12.88.2 Constructor & Destructor Documentation

12.88.2.1 DictPrinter()

`gdcM::DictPrinter::DictPrinter ()`

12.88.2.2 ~DictPrinter()

`gdcM::DictPrinter::~~DictPrinter ()` [default]

12.88.3 Member Function Documentation

12.88.3.1 Print()

`void gdcM::DictPrinter::Print (`
 `std::ostream & os)`

12.88.3.2 PrintDataElement2()

`void gdcM::DictPrinter::PrintDataElement2 (`
 `std::ostream & os,`
 `const DataSet & ds,`
 `const DataElement & ide)` [protected]

12.88.3.3 PrintDataSet2()

`void gdcM::DictPrinter::PrintDataSet2 (`
 `std::ostream & os,`
 `const DataSet & ds)` [protected]

The documentation for this class was generated from the following file:

- [gdcMDictPrinter.h](#)

12.89 gdcM::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load).

```
#include <gdcMDicts.h>
```

Public Member Functions

- [Dicts](#) ()
- [Dicts](#) (const [Dicts](#) &_val)=delete
- [~Dicts](#) ()
- const [CSAHeaderDict](#) & [GetCSAHeaderDict](#) () const
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag, const char *owner=nullptr) const
- [THREAD SAFE.](#)
- [PrivateDict](#) & [GetPrivateDict](#) ()
- const [PrivateDict](#) & [GetPrivateDict](#) () const
- const [Dict](#) & [GetPublicDict](#) () const
- bool [IsEmpty](#) () const
- [Dicts](#) & [operator=](#) (const [Dicts](#) &_val)=delete

Protected Types

- enum [ConstructorType](#) {
[PHILIPS](#) ,
[GEMS](#) ,
[SIEMENS](#) }

Protected Member Functions

- void [LoadDefaults](#) ()

Static Protected Member Functions

- static const char * [GetConstructorString](#) ([ConstructorType](#) type)

Friends

- class [Global](#)
- std::ostream & [operator<<](#) (std::ostream &_os, const [Dicts](#) &d)

12.89.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load).

Note

bla

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

12.89.2 Member Enumeration Documentation

12.89.2.1 ConstructorType

enum [gdcm::Dicts::ConstructorType](#) [protected]

Enumerator

PHILIPS	
GEMS	
SIEMENS	

12.89.3 Constructor & Destructor Documentation

12.89.3.1 Dicts() [1/2]

[gdcm::Dicts::Dicts](#) ()

Referenced by [Dicts\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.89.3.2 ~Dicts()

[gdcm::Dicts::~~Dicts](#) ()

12.89.3.3 Dicts() [2/2]

[gdcm::Dicts::Dicts](#) (
 const Dicts & __val) [delete]

References [Dicts\(\)](#).

12.89.4 Member Function Documentation

12.89.4.1 GetConstructorString()

```
const char * gdcmm::Dicts::GetConstructorString (
    ConstructorType type) [static], [protected]
```

12.89.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict & gdcmm::Dicts::GetCSAHeaderDict () const
```

Examples

[MrProtocol.cxx](#).

12.89.4.3 GetDictEntry() [1/2]

```
const DictEntry & gdcmm::Dicts::GetDictEntry (
    const PrivateTag & tag) const
```

12.89.4.4 GetDictEntry() [2/2]

```
const DictEntry & gdcmm::Dicts::GetDictEntry (
    const Tag & tag,
    const char * owner = nullptr) const
```

THREAD SAFE.

works for both public and private dicts: owner is null for public dict

Warning

owner need to be set to appropriate owner for call to work. see

Examples

[PublicDict.cxx](#), and [TraverseModules.cxx](#).

12.89.4.5 GetPrivateDict() [1/2]

```
PrivateDict & gdcmm::Dicts::GetPrivateDict ()
```

12.89.4.6 GetPrivateDict() [2/2]

const [PrivateDict](#) & gdcmm::Dicts::GetPrivateDict () const

12.89.4.7 GetPublicDict()

const [Dict](#) & gdcmm::Dicts::GetPublicDict () const

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

Referenced by [IsEmpty\(\)](#).

12.89.4.8 IsEmpty()

bool gdcmm::Dicts::IsEmpty () const [inline]

References [GetPublicDict\(\)](#).

12.89.4.9 LoadDefaults()

void gdcmm::Dicts::LoadDefaults () [protected]

12.89.4.10 operator=()

[Dicts](#) & gdcmm::Dicts::operator= (
const [Dicts](#) & __val) [delete]

References [Dicts\(\)](#).

12.89.5 Friends And Related Symbol Documentation

12.89.5.1 Global

friend class Global [friend]

References [Global](#).

Referenced by [Global](#).

12.89.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Dicts & d) [friend]
```

References [Dicts\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmDicts.h](#)

12.90 gdcm::network::DIMSE Class Reference

[DIMSE](#).

```
#include <gdcmDIMSE.h>
```

Public Types

- enum [CommandTypes](#) {
 - [C_STORE_RQ](#) = 0x0001 ,
 - [C_STORE_RSP](#) = 0x8001 ,
 - [C_GET_RQ](#) = 0x0010 ,
 - [C_GET_RSP](#) = 0x8010 ,
 - [C_FIND_RQ](#) = 0x0020 ,
 - [C_FIND_RSP](#) = 0x8020 ,
 - [C_MOVE_RQ](#) = 0x0021 ,
 - [C_MOVE_RSP](#) = 0x8021 ,
 - [C_ECHO_RQ](#) = 0x0030 ,
 - [C_ECHO_RSP](#) = 0x8030 ,
 - [N_EVENT_REPORT_RQ](#) = 0x0100 ,
 - [N_EVENT_REPORT_RSP](#) = 0x8100 ,
 - [N_GET_RQ](#) = 0x0110 ,
 - [N_GET_RSP](#) = 0x8110 ,
 - [N_SET_RQ](#) = 0x0120 ,
 - [N_SET_RSP](#) = 0x8120 ,
 - [N_ACTION_RQ](#) = 0x0130 ,
 - [N_ACTION_RSP](#) = 0x8130 ,
 - [N_CREATE_RQ](#) = 0x0140 ,
 - [N_CREATE_RSP](#) = 0x8140 ,
 - [N_DELETE_RQ](#) = 0x0150 ,
 - [N_DELETE_RSP](#) = 0x8150 ,
 - [C_CANCEL_RQ](#) = 0x0FFF }

12.90.1 Detailed Description

[DIMSE](#).

PS 3.7 - 2009 Annex E [Command](#) Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS [Table E.1-1](#) COMMAND FIELDS (PART 1)

12.90.2 Member Enumeration Documentation

12.90.2.1 CommandTypes

enum [gdcm::network::DIMSE::CommandTypes](#)

Enumerator

C_STORE_RQ	
C_STORE_RSP	
C_GET_RQ	
C_GET_RSP	
C_FIND_RQ	
C_FIND_RSP	
C_MOVE_RQ	
C_MOVE_RSP	
C_ECHO_RQ	
C_ECHO_RSP	
N_EVENT_REPORT_RQ	
N_EVENT_REPORT_RSP	
N_GET_RQ	
N_GET_RSP	
N_SET_RQ	
N_SET_RSP	
N_ACTION_RQ	
N_ACTION_RSP	
N_CREATE_RQ	
N_CREATE_RSP	
N_DELETE_RQ	
N_DELETE_RSP	
C_CANCEL_RQ	

The documentation for this class was generated from the following file:

- [gdcmDIMSE.h](#)

12.91 gdcm::DirectionCosines Class Reference

class to handle [DirectionCosines](#)

```
#include <gdcmDirectionCosines.h>
```

Public Member Functions

- [DirectionCosines](#) ()
- [DirectionCosines](#) (const double dircos[6])
- [~DirectionCosines](#) ()=default
- double [ComputeDistAlongNormal](#) (const double ipp[3]) const
Compute the distance along the normal.
- void [Cross](#) (double z[3]) const
Compute Cross product.
- double [CrossDot](#) ([DirectionCosines](#) const &dc) const
Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.
- double [Dot](#) () const
Compute Dot.
- bool [IsValid](#) () const
Return whether or not this is a valid direction cosines.
- void [Normalize](#) ()
Normalize in-place.
- [operator const double *](#) () const
Make the class behave like a const double *.
- void [Print](#) (std::ostream &) const
Print.
- bool [SetFromString](#) (const char *str)

Static Public Member Functions

- static double [Dot](#) (const double x[3], const double y[3])
Compute Dot.
- static double [Norm](#) (const double v[3])
Return norm of the vector.
- static void [Normalize](#) (double v[3])
Normalize in-place.

12.91.1 Detailed Description

class to handle [DirectionCosines](#)

Examples

[DiscriminateVolume.cxx](#).

12.91.2 Constructor & Destructor Documentation

12.91.2.1 [DirectionCosines](#)() [1/2]

`gdcmm::DirectionCosines::DirectionCosines ()`

Referenced by [CrossDot\(\)](#).

12.91.2.2 DirectionCosines() [2/2]

```
gdcM::DirectionCosines::DirectionCosines (
    const double dircos[6])
```

12.91.2.3 ~DirectionCosines()

```
gdcM::DirectionCosines::~~DirectionCosines () [default]
```

12.91.3 Member Function Documentation

12.91.3.1 ComputeDistAlongNormal()

```
double gdcM::DirectionCosines::ComputeDistAlongNormal (
    const double ipp[3]) const
```

Compute the distance along the normal.

12.91.3.2 Cross()

```
void gdcM::DirectionCosines::Cross (
    double z[3]) const
```

Compute Cross product.

12.91.3.3 CrossDot()

```
double gdcM::DirectionCosines::CrossDot (
    DirectionCosines const & dc) const
```

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

Examples

[DiscriminateVolume.cxx](#).

References [DirectionCosines\(\)](#).

12.91.3.4 Dot() [1/2]

```
double gdcM::DirectionCosines::Dot () const
```

Compute Dot.

12.91.3.5 Dot() [2/2]

```
double gdcM::DirectionCosines::Dot (  
    const double x[3],  
    const double y[3]) [static]
```

Compute Dot.

12.91.3.6 IsValid()

```
bool gdcM::DirectionCosines::IsValid () const
```

Return whether or not this is a valid direction cosines.

12.91.3.7 Norm()

```
double gdcM::DirectionCosines::Norm (  
    const double v[3]) [static]
```

Return norm of the vector.

12.91.3.8 Normalize() [1/2]

```
void gdcM::DirectionCosines::Normalize ()
```

Normalize in-place.

12.91.3.9 Normalize() [2/2]

```
void gdcM::DirectionCosines::Normalize (  
    double v[3]) [static]
```

Normalize in-place.

12.91.3.10 operator const double *()

```
gdcM::DirectionCosines::operator const double * () const [inline]
```

Make the class behave like a const double *.

12.91.3.11 Print()

```
void gdcM::DirectionCosines::Print (  
    std::ostream & ) const
```

Print.

12.91.3.12 SetFromString()

```
bool gdcmm::DirectionCosines::SetFromString (  
    const char * str)
```

Initialize from string str. It requires 6 floating point separated by a backslash character.

Examples

[DiscriminateVolume.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmDirectionCosines.h](#)

12.92 gdcmm::Directory Class Reference

Class for manipulation directories.

```
#include <gdcmDirectory.h>
```

Public Types

- typedef std::vector< [FilenameType](#) > [FilenamesType](#)
- typedef std::string [FilenameType](#)

Public Member Functions

- [Directory](#) ()=default
- [~Directory](#) ()=default
- [FilenamesType](#) const & [GetDirectories](#) () const
Return the Directories traversed.
- [FilenamesType](#) const & [GetFilenames](#) () const
Set/Get the file names within the directory.
- [FilenameType](#) const & [GetToplevel](#) () const
Get the name of the toplevel directory.
- unsigned int [Load](#) ([FilenameType](#) const &name, bool recursive=false)
- void [Print](#) (std::ostream &os=std::cout) const
Print.

Protected Member Functions

- unsigned int [Explore](#) ([FilenameType](#) const &name, bool recursive)
Return number of file found when 'recursive'ly exploring directory name.

Friends

- `std::ostream & operator<< (std::ostream &_os, const Directory &d)`

12.92.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directories: basically traversing directories and harvesting files

will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')

Since python or C# provide there own equivalent implementation, in which case [gdcm::Directory](#) does not make much sense.

Examples

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [ExplicitLittleEndian.cs](#), [GenerateDICOMDIR.cs](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), and [threadgdcm.cxx](#).

12.92.2 Member Typedef Documentation

12.92.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

Examples

[CStoreQtProgress.cxx](#), [DumpVisusChange.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [SimpleScanner.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcm.cxx](#).

12.92.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

12.92.3 Constructor & Destructor Documentation

12.92.3.1 Directory()

```
gdcm::Directory::Directory () [default]
```

Referenced by [operator<<](#).

12.92.3.2 ~Directory()

`gdcmm::Directory::~~Directory ()` [default]

12.92.4 Member Function Documentation

12.92.4.1 Explore()

`unsigned int gdcmm::Directory::Explore (`
 [FilenameType](#) const & name,
 bool recursive) [protected]

Return number of file found when 'recursive'ly exploring directory name.

12.92.4.2 GetDirectories()

[FilenameType](#) const & `gdcmm::Directory::GetDirectories ()` const [inline]

Return the Directories traversed.

12.92.4.3 GetFilenames()

[FilenameType](#) const & `gdcmm::Directory::GetFilenames ()` const [inline]

Set/Get the file names within the directory.

Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#),
[DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [ExplicitLittleEndian.cs](#), [GenerateDICOMDIR.cs](#),
[GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#),
[VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcmm.cxx](#).

References [gdcmm_assert](#).

12.92.4.4 GetToplevel()

[FilenameType](#) const & `gdcmm::Directory::GetToplevel ()` const [inline]

Get the name of the toplevel directory.

12.92.4.5 Load()

```
unsigned int gdcmm::Directory::Load (  
    FilenameType const & name,  
    bool recursive = false)
```

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples

[ClinicalTrialIdentificationWorkflow.cs](#), [DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#),
[DumpToSQLITE3.cxx](#), [DumpVisusChange.cxx](#), [ExplicitLittleEndian.cs](#), [GenerateDICOMDIR.cs](#),
[GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [ScanDirectory.cs](#), [SortImage.cxx](#), [StandardizeFiles.cs](#),
[VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [threadgdcmm.cxx](#).

12.92.4.6 Print()

```
void gdcmm::Directory::Print (  
    std::ostream & os = std::cout) const
```

Print.

Examples

[SortImage.cxx](#).

Referenced by [operator<<](#).

12.92.5 Friends And Related Symbol Documentation

12.92.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const Directory & d) [friend]
```

References [Directory\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmDirectory.h](#)

12.93 gdcm::DirectoryHelper Class Reference

[DirectoryHelper](#).

```
#include <gdcmDirectoryHelper.h>
```

Static Public Member Functions

- static [Directory::FileNamesType](#) [GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FileNamesType](#) [GetFileNamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FileNamesType](#) [GetMRImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FileNamesType](#) [GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FileNamesType](#) [GetSeriesUIDsBySOPClassUID](#) (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string [GetSOPClassUID](#) (const std::vector< [DataSet](#) > &inDS)
- static std::string [GetStringValueFromTag](#) (const [Tag](#) &t, const [DataSet](#) &ds)
- static std::vector< [DataSet](#) > [LoadImageFromFiles](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [RetrieveSOPInstanceUIDFromIndex](#) (int inIndex, const std::vector< [DataSet](#) > &inDS)
- static std::string [RetrieveSOPInstanceUIDFromZPosition](#) (double inZPos, const std::vector< [DataSet](#) > &inDS)

12.93.1 Detailed Description

[DirectoryHelper](#).

this class is designed to help mitigate some of the commonly performed operations on directories. namely: 1) the ability to determine the number of series in a directory by what type of series is present 2) the ability to find all ct series in a directory 3) the ability to find all mr series in a directory 4) to load a set of DataSets from a series that's already been sorted by the IPP sorter 5) For rtstruct stuff, you need to know the sopinstanceuid of each z plane, so there's a retrieval function for that 6) then a few other functions for rtstruct writeouts

12.93.2 Member Function Documentation

12.93.2.1 GetCTImageSeriesUIDs()

[Directory::FileNamesType](#) gdcm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & inDirectory) [static]

12.93.2.2 GetFilenamesFromSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (  
    const std::string & inDirectory,  
    const std::string & inSeriesUID) [static]
```

Examples

[GenerateRTSTRUCT.cxx](#).

12.93.2.3 GetFrameOfReference()

```
std::string gdcm::DirectoryHelper::GetFrameOfReference (  
    const std::vector< DataSet > & inDS) [static]
```

12.93.2.4 GetMRImageSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (  
    const std::string & inDirectory) [static]
```

12.93.2.5 GetRTStructSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (  
    const std::string & inDirectory) [static]
```

Examples

[GenerateRTSTRUCT.cxx](#).

12.93.2.6 GetSeriesUIDsBySOPClassUID()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (  
    const std::string & inDirectory,  
    const std::string & inSOPClassUID) [static]
```

12.93.2.7 GetSOPClassUID()

```
std::string gdcm::DirectoryHelper::GetSOPClassUID (  
    const std::vector< DataSet > & inDS) [static]
```

12.93.2.8 GetStringValueFromTag()

```
std::string gdcmm::DirectoryHelper::GetStringValueFromTag (
    const Tag & t,
    const DataSet & ds) [static]
```

12.93.2.9 LoadImageFromFiles()

```
std::vector< DataSet > gdcmm::DirectoryHelper::LoadImageFromFiles (
    const std::string & inDirectory,
    const std::string & inSeriesUID) [static]
```

12.93.2.10 RetrieveSOPInstanceUIDFromIndex()

```
std::string gdcmm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
    int inIndex,
    const std::vector< DataSet > & inDS) [static]
```

12.93.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
std::string gdcmm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
    double inZPos,
    const std::vector< DataSet > & inDS) [static]
```

The documentation for this class was generated from the following file:

- [gdcmmDirectoryHelper.h](#)

12.94 gdcmm::DPath Class Reference

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation [https://groups.google.com/g/comp.↵ protocols.dicom/c/IyIH0IOBMPA](https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA)

```
#include <gdcmmDPath.h>
```

Public Member Functions

- [DPath](#) ()
- [~DPath](#) ()
- bool [ConstructFromString](#) (const char *path)
- bool [Match](#) ([DPath](#) const &other) const
Return whether or not 'other' match the template [DPath](#).
- bool [operator<](#) (const [DPath](#) &rhs) const
- void [Print](#) (std::ostream &) const

Static Public Member Functions

- static bool [IsValid](#) (const char *path)
Return if path is valid or not.

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [DPath](#) &_val)

12.94.1 Detailed Description

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation [https://groups.google.com/g/comp.↵ protocols.dicom/c/IyIH0IOBMPA](https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA)

12.94.2 Constructor & Destructor Documentation

12.94.2.1 DPath()

gdcm::DPath::DPath ()

Referenced by [Match\(\)](#), [operator<\(\)](#), and [operator<<](#).

12.94.2.2 ~DPath()

gdcm::DPath::~~DPath ()

12.94.3 Member Function Documentation

12.94.3.1 ConstructFromString()

bool gdcm::DPath::ConstructFromString (
 const char * path)

Examples

[Cleaner.cs](#).

12.94.3.2 IsValid()

bool gdcm::DPath::IsValid (
 const char * path) [static]

Return if path is valid or not.

12.94.3.3 Match()

```
bool gdcmm::DPath::Match (  
    DPath const & other) const
```

Return whether or not 'other' match the template [DPath](#).

References [DPath\(\)](#).

12.94.3.4 operator<()

```
bool gdcmm::DPath::operator< (  
    const DPath & rhs) const
```

References [DPath\(\)](#).

12.94.3.5 Print()

```
void gdcmm::DPath::Print (  
    std::ostream & ) const
```

12.94.4 Friends And Related Symbol Documentation

12.94.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const DPath & __val) [friend]
```

References [DPath\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmDPath.h](#)

12.95 gdcmm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmmDummyValueGenerator.h>
```

Static Public Member Functions

- static const char * [Generate](#) (const char *input)

12.95.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

12.95.2 Member Function Documentation

12.95.2.1 Generate()

```
const char * gdcm::DummyValueGenerator::Generate (  
    const char * input) [static]
```

Generate a dummy value from an input value. This is guarantee to always return the same output value when input is identical. Return an array of bytes that can be used for anonymization purpose, return NULL on error NOT THREAD SAFE

The documentation for this class was generated from the following file:

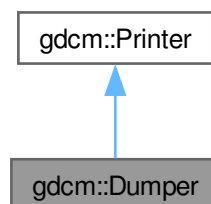
- [gdcmDummyValueGenerator.h](#)

12.96 gdcm::Dumper Class Reference

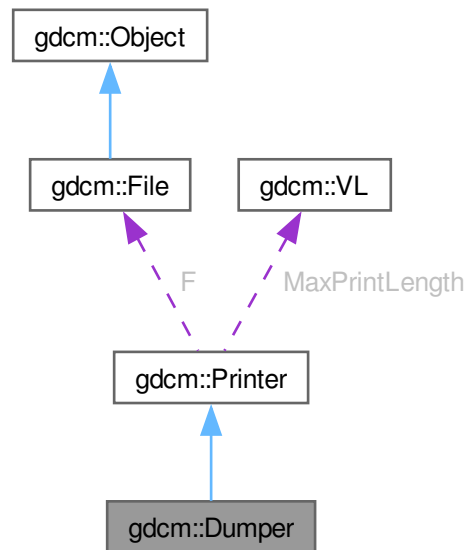
[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for `gdcM::Dumper`:



Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()=default

Public Member Functions inherited from [gdcM::Printer](#)

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles GetPrintStyle](#) () const
Get PrintStyle value.
- void [Print](#) (std::ostream &os)
Print.
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &s="")
Print an individual dataset.
- void [SetColor](#) (bool c)
Set color mode or not.
- void [SetFile](#) ([File](#) const &f)
Set file.
- void [SetStyle](#) ([PrintStyles](#) ps)
Set PrintStyle value.

Additional Inherited Members

Public Types inherited from [gdcm::Printer](#)

- enum [PrintStyles](#) {
 [VERBOSE_STYLE](#) = 0 ,
 [CONDENSED_STYLE](#) ,
 [XML](#) ,
 [CXX](#) }

Protected Member Functions inherited from [gdcm::Printer](#)

- [VR PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) *sqi, std::ostream &os, std::string const &indent)

Protected Attributes inherited from [gdcm::Printer](#)

- const [File](#) * F
- [VL MaxPrintLength](#)
- [PrintStyles PrintStyle](#)

12.96.1 Detailed Description

[Codec](#) class.

Note

Use it to simply dump value read from the file. No interpretation is done. But it is real fast ! Almost no overhead

12.96.2 Constructor & Destructor Documentation

12.96.2.1 Dumper()

[gdcm::Dumper::Dumper](#) () [inline]

References [gdcm::Printer::CONDENSED_STYLE](#), and [gdcm::Printer::PrintStyle](#).

12.96.2.2 ~Dumper()

[gdcm::Dumper::~Dumper](#) () [default]

The documentation for this class was generated from the following file:

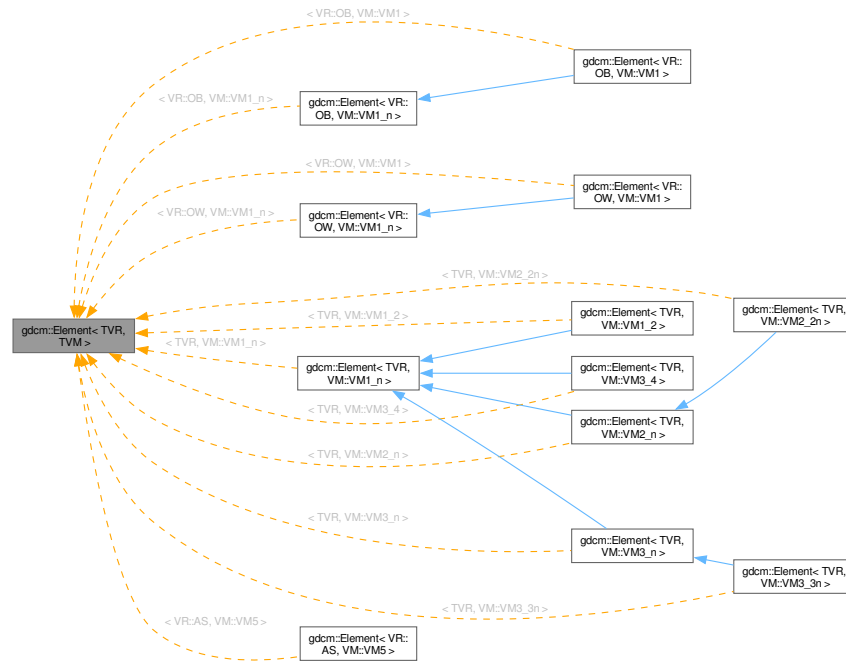
- [gdcmDumper.h](#)

12.97 gdcm::Element< TVR, TVM > Class Template Reference

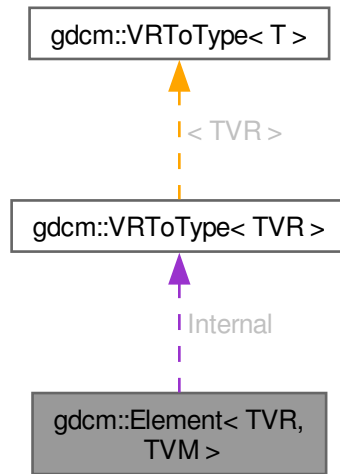
[Element](#) class.

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, TVM >`:



Collaboration diagram for gdcm::Element< TVR, TVM >:



Public Types

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement` `GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- const `VRToType< TVR >::Type` * `GetValues` () const
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const
- void `Print` (std::ostream &_os) const
- void `Read` (std::istream &_is)
- void `Set` (`Value` const &v)
- void `SetFromDataElement` (`DataElement` const &de)
- void `SetValue` (typename `VRToType< TVR >::Type` v, unsigned int idx=0)
- void `Write` (std::ostream &_os) const

Static Public Member Functions

- static `VM` `GetVM` ()
- static `VR` `GetVR` ()

Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetNoSwap](#) (Value const &v)

12.97.1 Detailed Description

```
template<long long TVR, int TVM>
class gdcmm::Element< TVR, TVM >
```

[Element](#) class.

Note

TODO

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GetSubSequenceData.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.97.2 Member Typedef Documentation

12.97.2.1 Type

```
template<long long TVR, int TVM>
typedef VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Type
```

12.97.3 Member Function Documentation

12.97.3.1 GetAsDataElement()

```
template<long long TVR, int TVM>
DataElement gdcmm::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

Examples

[Extracting_All_Resolution.cxx](#), and [Fake_Image_Using_Stream_Image_Writer.cxx](#).

References [gdcmm_assert](#), [GetLength\(\)](#), [gdcmm::DataElement::GetVR\(\)](#), [GetVR\(\)](#), [Internal](#), [gdcmm::DataElement::SetByteValue\(\)](#), [gdcmm::DataElement::SetVR\(\)](#), [gdcmm::VR::SQ](#), [gdcmm::VR::UI](#), [gdcmm::VR::VRASCII](#), and [Write\(\)](#).

12.97.3.2 GetLength()

```
template<long long TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength () const [inline]
```

Examples

[DumpGEMSMovieGroup.cxx](#), and [GetSubSequenceData.cxx](#).

Referenced by [GetAsDataElement\(\)](#), [Read\(\)](#), [Set\(\)](#), [SetNoSwap\(\)](#), and [Write\(\)](#).

12.97.3.3 GetValue() [1/2]

```
template<long long TVR, int TVM>
VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) [inline]
```

References [gdcm__assert](#), and [Internal](#).

12.97.3.4 GetValue() [2/2]

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#),
[GetSubSequenceData.cxx](#), and [csa2img.cxx](#).

References [gdcm__assert](#), and [Internal](#).

Referenced by [operator\[\]\(\)](#).

12.97.3.5 GetValues()

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const [inline]
```

References [Internal](#).

12.97.3.6 GetVM()

```
template<long long TVR, int TVM>
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.97.3.7 GetVR()

```
template<long long TVR, int TVM>
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

Referenced by [GetAsDataElement\(\)](#).

12.97.3.8 operator[]()

```
template<long long TVR, int TVM>
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

References [GetValue\(\)](#).

12.97.3.9 Print()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
    std::ostream & __os) const [inline]
```

Examples

[DumpGEMSMovieGroup.cxx](#).

References [Internal](#).

12.97.3.10 Read()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
    std::istream & __is) [inline]
```

References [GetLength\(\)](#), [Internal](#), and [Read\(\)](#).

Referenced by [Read\(\)](#), and [Set\(\)](#).

12.97.3.11 Set()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

Examples

[csa2img.cxx](#).

References [gdcm::ByteValue::GetLength\(\)](#), [GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), [Internal](#), and [Read\(\)](#).

Referenced by [SetFromDataElement\(\)](#).

12.97.3.12 SetFromDataElement()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, TVM > const & de) [inline]
```

Examples

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

References [gdcmm::DataElement::GetByteValue\(\)](#), [gdcmm::DataElement::GetValue\(\)](#), [gdcmm::DataElement::GetVR\(\)](#), [gdcmm::VR::INVALID](#), [Set\(\)](#), [SetNoSwap\(\)](#), and [gdcmm::VR::UN](#).

12.97.3.13 SetNoSwap()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

References [gdcmm_assert](#), [gdcmm::ByteValue::GetLength\(\)](#), [GetLength\(\)](#), [gdcmm::ByteValue::GetPointer\(\)](#), and [Internal](#).

Referenced by [SetFromDataElement\(\)](#).

12.97.3.14 SetValue()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

Examples

[Extracting_All_Resolution.cxx](#), and [Fake_Image_Using_Stream_Image_Writer.cxx](#).

References [gdcmm_assert](#), and [Internal](#).

12.97.3.15 Write()

```
template<long long TVR, int TVM>
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & __os) const [inline]
```

References [GetLength\(\)](#), [Internal](#), and [Write\(\)](#).

Referenced by [GetAsDataElement\(\)](#), and [Write\(\)](#).

12.97.4 Member Data Documentation

12.97.4.1 Internal

```
template<long long TVR, int TVM>
```

```
VRToType<TVR>::Type gdcElement< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by [GetAsDataElement\(\)](#), [GetValue\(\)](#), [GetValue\(\)](#), [GetValues\(\)](#), [Print\(\)](#), [Read\(\)](#), [Set\(\)](#), [SetNoSwap\(\)](#), [SetValue\(\)](#), and [Write\(\)](#).

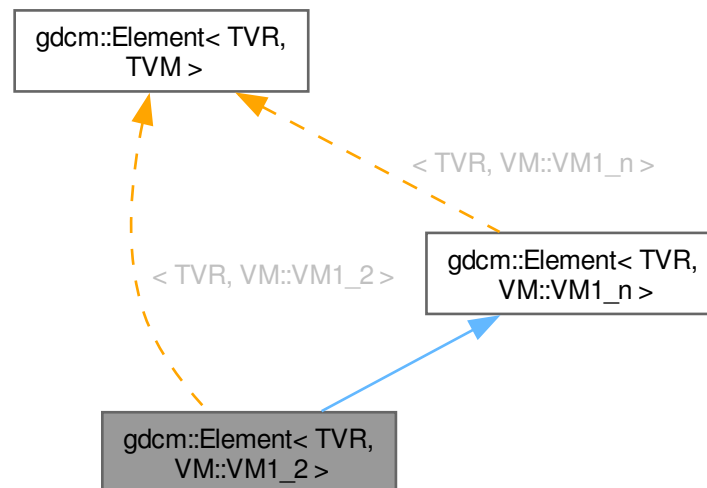
The documentation for this class was generated from the following file:

- [gdcElement.h](#)

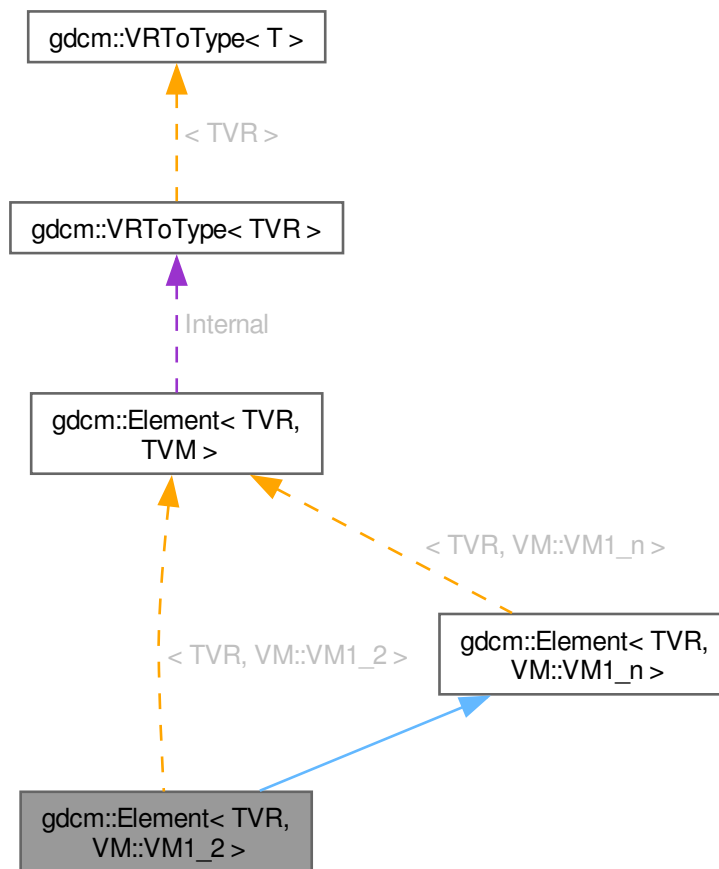
12.98 gdcElement< TVR, VM::VM1_2 > Class Template Reference

```
#include <gdcElement.h>
```

Inheritance diagram for gdcElement< TVR, VM::VM1_2 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1_2 >:



Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`

- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- [Element](#) ()
- [Element](#) (const Element &_val)
- [~Element](#) ()
- DataElement [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- Element & operator= (const Element &_val)
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetArray](#) (const Type *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcm::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.98.1 Member Typedef Documentation

12.98.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcm::Element< TVR, VM::VM1\_2 >::Parent
```

12.98.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.98.2 Member Function Documentation

12.98.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.98.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const [inline]
```

12.98.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.98.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const [inline]
```

12.98.2.5 GetVM()

```
VM gdcmm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.98.2.6 GetVR()

```
VR gdcmm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.98.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

12.98.2.8 Print()

```
void gdcmm::Element< TVR, TVM >::Print (
    std::ostream & _os) const [inline]
```

12.98.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & _is) [inline]
```

12.98.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

12.98.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, VM::VM1_2 > const & de) [inline]
```

12.98.2.12 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_2 >::SetLength (
    int len) [inline]
```

References [gdcmm::Element< TVR, VM::VM1_n >::SetLength\(\)](#).

12.98.2.13 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

12.98.2.14 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

12.98.2.15 Write()

```
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & _os) const [inline]
```

12.98.3 Member Data Documentation

12.98.3.1 Internal

```
VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

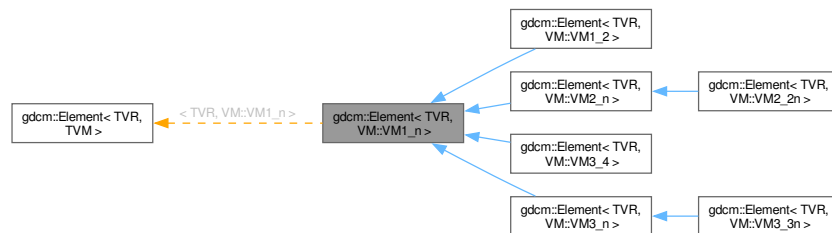
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

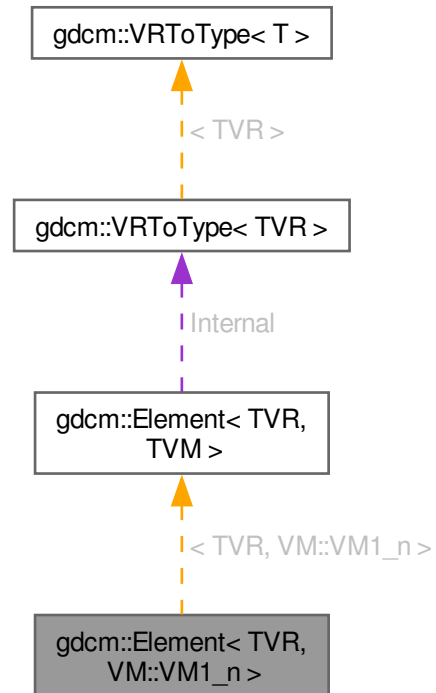
12.99 gdcmm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::Element< TVR, VM::VM1_n >:



Collaboration diagram for `gdcm::Element< TVR, VM::VM1_n >`:



Public Types

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `Element` ()
- `Element` (const `Element` &_val)
- `~Element` ()
- `DataElement` `GetAsDataElement` () const
- unsigned long `GetLength` () const
- `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- const `VRToType< TVR >::Type` * `GetValues` () const
- `Element` & `operator=` (const `Element` &_val)
- `VRToType< TVR >::Type` `operator[]` (unsigned int idx) const
- void `Print` (std::ostream &_os) const
- void `Read` (std::istream &_is)
- void `Set` (`Value` const &v)

- void [SetArray](#) (const [Type](#) *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

12.99.1 Member Typedef Documentation

12.99.1.1 Type

```
template<long long TVR>
typedef VRToType<TVR>::Type gdcmm::Element< TVR, VM::VM1\_n >::Type
```

12.99.2 Constructor & Destructor Documentation

12.99.2.1 Element() [1/2]

```
template<long long TVR>
gdcmm::Element< TVR, VM::VM1\_n >::Element () [inline], [explicit]
```

Referenced by [Element\(\)](#), and [operator=\(\)](#).

12.99.2.2 ~Element()

```
template<long long TVR>
gdcmm::Element< TVR, VM::VM1\_n >::~~Element () [inline]
```

12.99.2.3 Element() [2/2]

```
template<long long TVR>
gdcmm::Element< TVR, VM::VM1\_n >::Element (
    const Element< TVR, VM::VM1\_n > & _val) [inline]
```

References [Element\(\)](#).

12.99.3 Member Function Documentation

12.99.3.1 GetAsDataElement()

```
template<long long TVR>
DataElement gdcM::Element< TVR, VM::VM1_n >::GetAsDataElement () const [inline]
```

References [gdcM_assert](#), [GetLength\(\)](#), [gdcM::DataElement::GetVR\(\)](#), [GetVR\(\)](#), [gdcM::DataElement::SetByteValue\(\)](#), [gdcM::DataElement::SetVR\(\)](#), [gdcM::VR::SQ](#), [gdcM::VR::UI](#), [gdcM::VR::VRASCII](#), and [Write\(\)](#).

12.99.3.2 GetLength()

```
template<long long TVR>
unsigned long gdcM::Element< TVR, VM::VM1_n >::GetLength () const [inline]
```

Referenced by [GetAsDataElement\(\)](#), [Print\(\)](#), [Read\(\)](#), [Set\(\)](#), [SetNoSwap\(\)](#), [Write\(\)](#), and [WriteASCII\(\)](#).

12.99.3.3 GetValue() [1/2]

```
template<long long TVR>
VRToType< TVR >::Type & gdcM::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0) [inline]
```

12.99.3.4 GetValue() [2/2]

```
template<long long TVR>
const VRToType< TVR >::Type & gdcM::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0) const [inline]
```

References [gdcM_assert](#).

Referenced by [operator\[\]\(\)](#).

12.99.3.5 GetValues()

```
const VRToType< TVR >::Type * gdcM::Element< TVR, TVM >::GetValues () const [inline]
```

12.99.3.6 GetVM()

```
template<long long TVR>
VM gdcM::Element< TVR, VM::VM1_n >::GetVM () [inline], [static]
```

References [gdcM::VM::VM1_n](#).

12.99.3.7 GetVR()

```
template<long long TVR>
VR gdcmm::Element< TVR, VM::VM1_n >::GetVR () [inline], [static]
```

Referenced by [GetAsDataElement\(\)](#).

12.99.3.8 operator=()

```
template<long long TVR>
Element & gdcmm::Element< TVR, VM::VM1_n >::operator= (
    const Element< TVR, VM::VM1_n > & __val) [inline]
```

References [Element\(\)](#), and [SetArray\(\)](#).

12.99.3.9 operator[]()

```
template<long long TVR>
VRToType< TVR >::Type gdcmm::Element< TVR, VM::VM1_n >::operator[] (
    unsigned int idx) const [inline]
```

References [GetValue\(\)](#).

12.99.3.10 Print()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::Print (
    std::ostream & __os) const [inline]
```

References [gdcmm_assert](#), and [GetLength\(\)](#).

12.99.3.11 Read()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM1_n >::Read (
    std::istream & __is) [inline]
```

References [GetLength\(\)](#), and [Read\(\)](#).

Referenced by [Read\(\)](#), and [Set\(\)](#).

12.99.3.12 Set()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
    Value const & v) [inline]
```

References [gdcm_assert](#), [gdcm::ByteValue::GetLength\(\)](#), [GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), [gdcm::ByteValue::GetVoidPointer\(\)](#), [Read\(\)](#), [SetArray\(\)](#), and [gdcm::VR::VRBINARY](#).

Referenced by [SetFromDataElement\(\)](#).

12.99.3.13 SetArray()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
    const Type * array,
    unsigned long len,
    bool save = false) [inline]
```

References [gdcm_assert](#), and [SetLength\(\)](#).

Referenced by [operator=\(\)](#), [Set\(\)](#), and [SetNoSwap\(\)](#).

12.99.3.14 SetFromDataElement()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (
    DataElement< TVR, VM::VM1_n > const & de) [inline]
```

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::DataElement::GetValue\(\)](#), [gdcm::DataElement::GetVR\(\)](#), [gdcm::VR::INVALID](#), [Set\(\)](#), [SetNoSwap\(\)](#), and [gdcm::VR::UN](#).

12.99.3.15 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
    unsigned long len) [inline]
```

References [gdcm_assert](#).

Referenced by [SetArray\(\)](#), [gdcm::Element< TVR, VM::VM1_2 >::SetLength\(\)](#), [gdcm::Element< TVR, VM::VM2_n >::SetLength\(\)](#), [gdcm::Element< TVR, VM::VM3_4 >::SetLength\(\)](#), and [gdcm::Element< TVR, VM::VM3_n >::SetLength\(\)](#).

12.99.3.16 SetNoSwap()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
    Value const & v) [inline], [protected]
```

References [gdcm_assert](#), [gdcm::ByteValue::GetLength\(\)](#), [GetLength\(\)](#), [gdcm::ByteValue::GetPointer\(\)](#), [SetArray\(\)](#), and [gdcm::VR::VRBINARY](#).

Referenced by [SetFromDataElement\(\)](#).

12.99.3.17 SetValue()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

References [gdcm_assert](#).

12.99.3.18 Write()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Write (
    std::ostream & _os) const [inline]
```

References [GetLength\(\)](#), and [Write\(\)](#).

Referenced by [GetAsDataElement\(\)](#), and [Write\(\)](#).

12.99.3.19 WriteASCII()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (
    std::ostream & os) const [inline]
```

References [GetLength\(\)](#).

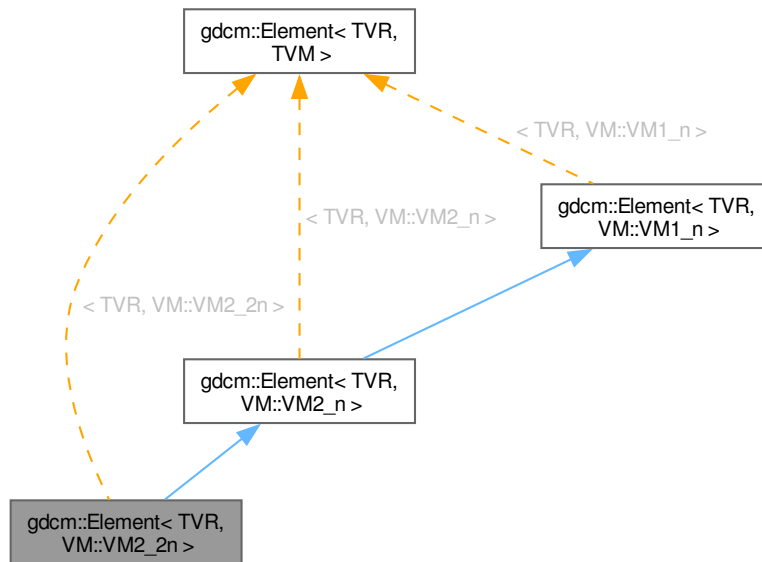
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

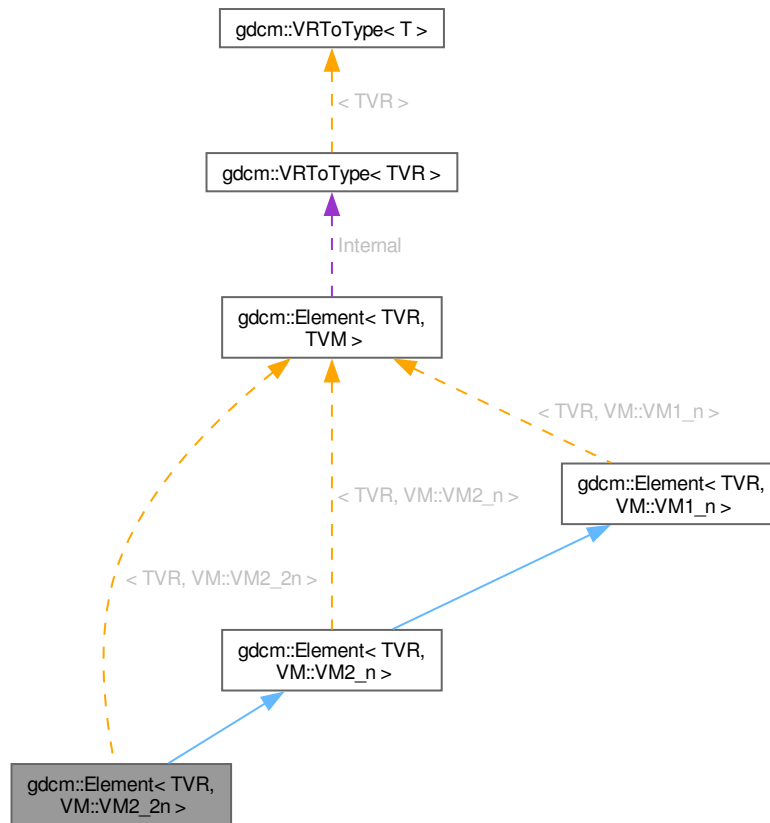
12.100 gdcElement< TVR, VM::VM2_2n > Class Template Reference

```
#include <gdcElement.h>
```

Inheritance diagram for gdcElement< TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_2n >:



Public Types

- typedef `Element< TVR, VM::VM2_n >` `Parent`
- typedef `VRTToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM2_n >`

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRTToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRTToType< TVR >::Type` `Type`

Public Member Functions

- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM2_n >

- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- [Element](#) ()
- [Element](#) (const [Element](#) &_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [Element](#) & [operator=](#) (const [Element](#) &_val)
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetArray](#) (const [Type](#) *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Element< TVR, VM::VM2_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Element< TVR, VM::VM1_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Public Attributes inherited from [gdcmm::Element< TVR, VM::VM2_n >](#)

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcmm::Element< TVR, VM::VM2_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcmm::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.100.1 Member Typedef Documentation

12.100.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM2_n> gdcmm::Element< TVR, VM::VM2_2n >::Parent
```

12.100.1.2 Type

```
typedef VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Type
```

12.100.2 Member Function Documentation

12.100.2.1 GetAsDataElement()

```
DataElement gdcmm::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.100.2.2 GetLength()

```
unsigned long gdcmm::Element< TVR, TVM >::GetLength () const [inline]
```

12.100.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcmm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.100.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcmm::Element< TVR, TVM >::GetValues () const [inline]
```

12.100.2.5 GetVM()

```
VM gdcmm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.100.2.6 GetVR()

```
VR gdcmm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.100.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

12.100.2.8 Print()

```
void gdcmm::Element< TVR, TVM >::Print (
    std::ostream & __os) const [inline]
```

12.100.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & __is) [inline]
```

12.100.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

12.100.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, VM::VM2_2n > const & de) [inline]
```

12.100.2.12 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM2_2n >::SetLength (
    int len) [inline]
```

References [gdcmm::Element< TVR, VM::VM2_n >::SetLength\(\)](#).

12.100.2.13 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

12.100.2.14 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

12.100.2.15 Write()

```
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & __os) const [inline]
```

12.100.3 Member Data Documentation

12.100.3.1 Internal

[VRToType<TVR>::Type](#) [gdcm::Element< TVR, TVM >::Internal](#)[[VMToLength< TVM >::Length](#)]

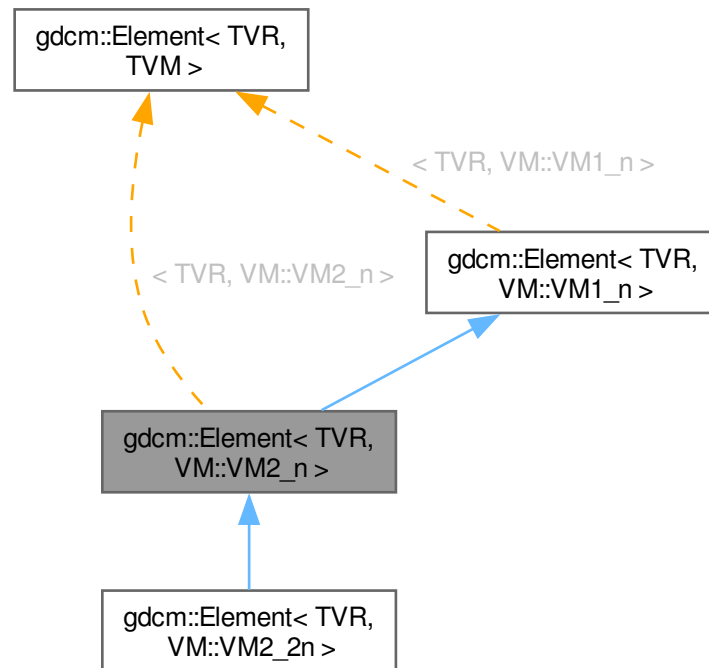
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

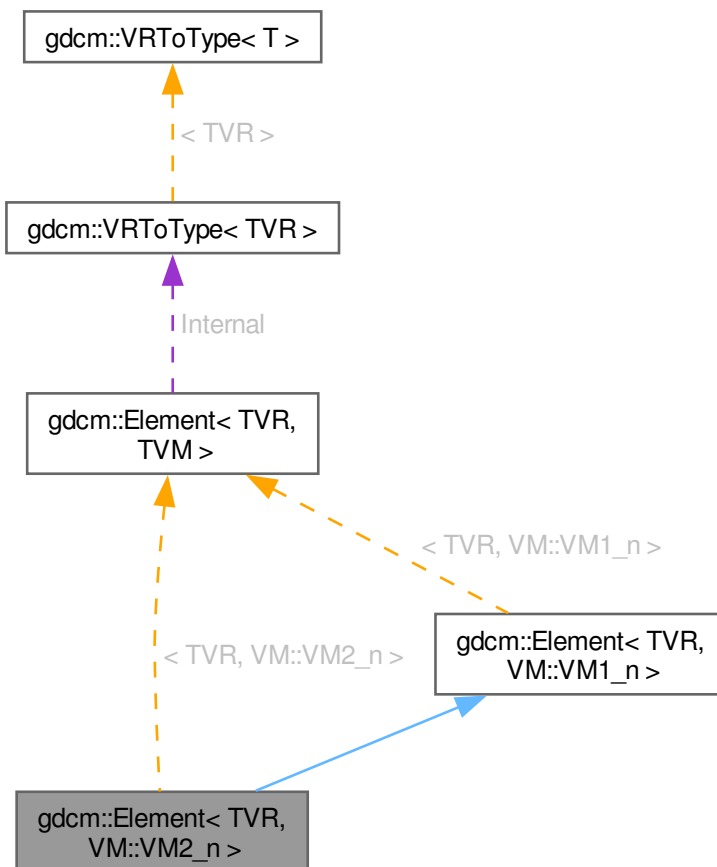
12.101 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM2_n >`:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_n >:



Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`

- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- [Element](#) ()
- [Element](#) (const Element &_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [Element](#) & operator= (const [Element](#) &_val)
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetArray](#) (const Type *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcmm::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.101.1 Member Typedef Documentation

12.101.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcmm::Element< TVR, VM::VM2\_n>::Parent
```

12.101.1.2 Type

```
typedef VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Type
```

12.101.2 Member Function Documentation

12.101.2.1 GetAsDataElement()

```
DataElement gdcmm::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.101.2.2 GetLength()

```
unsigned long gdcmm::Element< TVR, TVM >::GetLength () const [inline]
```

12.101.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcmm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.101.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcmm::Element< TVR, TVM >::GetValues () const [inline]
```

12.101.2.5 GetVM()

[VM](#) [gdcmm::Element](#)< TVR, TVM >::GetVM () [inline], [static]

12.101.2.6 GetVR()

[VR](#) [gdcmm::Element](#)< TVR, TVM >::GetVR () [inline], [static]

12.101.2.7 operator[]()

[VRToType](#)< TVR >::Type [gdcmm::Element](#)< TVR, TVM >::operator[] (unsigned int idx) const [inline]

12.101.2.8 Print()

void [gdcmm::Element](#)< TVR, TVM >::Print (std::ostream & __os) const [inline]

12.101.2.9 Read()

void [gdcmm::Element](#)< TVR, TVM >::Read (std::istream & __is) [inline]

12.101.2.10 Set()

void [gdcmm::Element](#)< TVR, TVM >::Set ([Value](#) const & v) [inline]

12.101.2.11 SetFromDataElement()

void [gdcmm::Element](#)< TVR, TVM >::SetFromDataElement ([DataElement](#)< TVR, [VM::VM2_n](#) > const & de) [inline]

12.101.2.12 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM2\_n >::SetLength (
    int len) [inline]
```

References [gdcmm::Element](#)< TVR, [VM::VM1_n](#) >::SetLength().

Referenced by [gdcmm::Element](#)< TVR, [VM::VM2_2n](#) >::SetLength().

12.101.2.13 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (  
    Value const & v)    [inline], [protected]
```

12.101.2.14 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (  
    typename VRToType< TVR >::Type v,  
    unsigned int idx = 0)    [inline]
```

12.101.2.15 Write()

```
void gdcmm::Element< TVR, TVM >::Write (  
    std::ostream & _os) const    [inline]
```

12.101.3 Member Data Documentation

12.101.3.1 Internal

[VRToType<TVR>::Type](#) [gdcmm::Element< TVR, TVM >::Internal](#)[[VMToLength< TVM >::Length](#)]

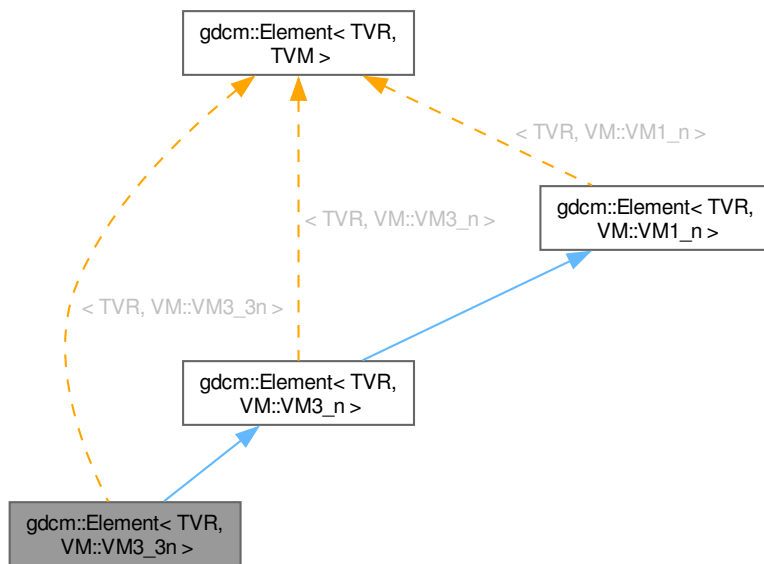
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

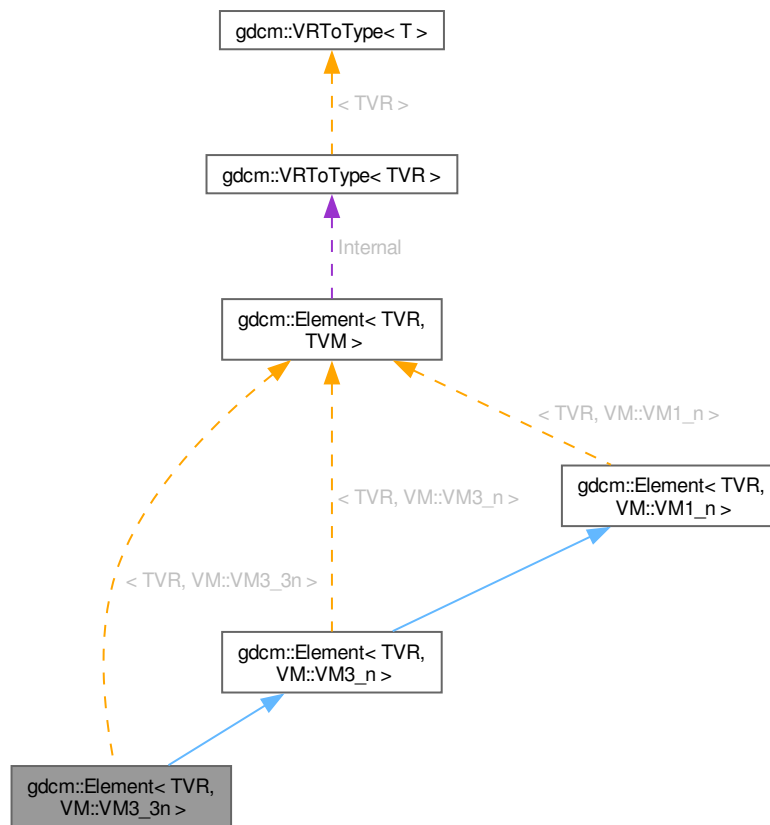
12.102 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for `gdc::Element< TVR, VM::VM3_3n >`:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_3n >:



Public Types

- typedef `Element< TVR, VM::VM3_n >` `Parent`
- typedef `VRTToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM3_n >`

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRTToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRTToType< TVR >::Type` `Type`

Public Member Functions

- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element< TVR, VM::VM3_n >](#)

- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1_n >](#)

- [Element](#) ()
- [Element](#) (const [Element](#) &_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [Element](#) & [operator=](#) (const [Element](#) &_val)
- [VRToType](#)< TVR >::Type [operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetArray](#) (const [Type](#) *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Element< TVR, VM::VM3_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcmm::Element< TVR, VM::VM1_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Public Attributes inherited from [gdcmm::Element< TVR, VM::VM3_n >](#)

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcmm::Element< TVR, VM::VM3_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcmm::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.102.1 Member Typedef Documentation

12.102.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM3_n> gdcmm::Element< TVR, VM::VM3_3n >::Parent
```

12.102.1.2 Type

```
typedef VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Type
```

12.102.2 Member Function Documentation

12.102.2.1 GetAsDataElement()

```
DataElement gdcmm::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.102.2.2 GetLength()

```
unsigned long gdcmm::Element< TVR, TVM >::GetLength () const [inline]
```

12.102.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcmm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.102.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcmm::Element< TVR, TVM >::GetValues () const [inline]
```

12.102.2.5 GetVM()

```
VM gdcmm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.102.2.6 GetVR()

```
VR gdcmm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.102.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

12.102.2.8 Print()

```
void gdcmm::Element< TVR, TVM >::Print (
    std::ostream & __os) const [inline]
```


12.102.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & __is) [inline]
```

12.102.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

12.102.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, VM::VM3_3n > const & de) [inline]
```

12.102.2.12 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM3_3n >::SetLength (
    int len) [inline]
```

References [gdcmm::Element< TVR, VM::VM3_n >::SetLength\(\)](#).

12.102.2.13 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

12.102.2.14 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

12.102.2.15 Write()

```
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & __os) const [inline]
```

12.102.3 Member Data Documentation

12.102.3.1 Internal

[VRToType<TVR>::Type](#) [gdcm::Element< TVR, TVM >::Internal](#)[[VMToLength< TVM >::Length](#)]

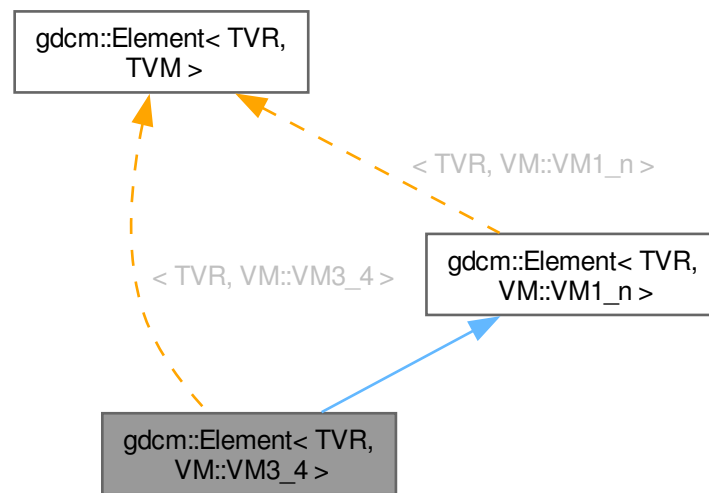
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

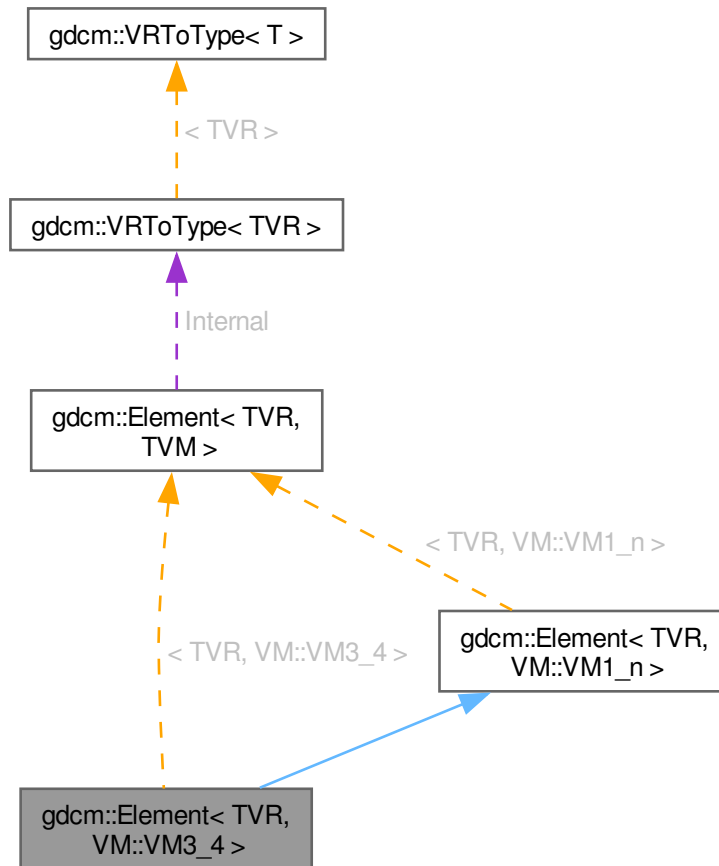
12.103 gdcm::Element< TVR, VM::VM3_4 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM3_4 >`:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_4 >:



Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`

- const [VRToType< TVR >::Type & GetValue](#) (unsigned int idx=0) const
- const [VRToType< TVR >::Type * GetValues](#) () const
- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1_n >](#)

- [Element](#) ()
- [Element](#) (const [Element](#) &_val)
- [~Element](#) ()
- [DataElement GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType< TVR >::Type & GetValue](#) (unsigned int idx=0)
- const [VRToType< TVR >::Type & GetValue](#) (unsigned int idx=0) const
- const [VRToType< TVR >::Type * GetValues](#) () const
- [Element & operator=](#) (const [Element](#) &_val)
- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetArray](#) (const [Type](#) *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element< TVR, VM::VM1_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [VRToType< TVR >::Type Internal \[VMToLength< TVM >::Length\]](#)

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcM::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.103.1 Member Typedef Documentation

12.103.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcM::Element< TVR, VM::VM3\_4 >::Parent
```

12.103.1.2 Type

```
typedef VRToType<TVR>::Type gdcM::Element< TVR, TVM >::Type
```

12.103.2 Member Function Documentation

12.103.2.1 GetAsDataElement()

```
DataElement gdcM::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.103.2.2 GetLength()

```
unsigned long gdcM::Element< TVR, TVM >::GetLength () const [inline]
```

12.103.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcM::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.103.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcM::Element< TVR, TVM >::GetValues () const [inline]
```

12.103.2.5 GetVM()

```
VM gdcmm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.103.2.6 GetVR()

```
VR gdcmm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.103.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

12.103.2.8 Print()

```
void gdcmm::Element< TVR, TVM >::Print (
    std::ostream & _os) const [inline]
```

12.103.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & _is) [inline]
```

12.103.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

12.103.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, VM::VM3_4 > const & de) [inline]
```

12.103.2.12 SetLength()

```
template<long long TVR>
void gdcmm::Element< TVR, VM::VM3_4 >::SetLength (
    int len) [inline]
```

References [gdcmm::Element< TVR, VM::VM1_n >::SetLength\(\)](#).

12.103.2.13 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v)    [inline], [protected]
```

12.103.2.14 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0)    [inline]
```

12.103.2.15 Write()

```
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & __os) const    [inline]
```

12.103.3 Member Data Documentation

12.103.3.1 Internal

[VRToType<TVR>::Type](#) [gdcmm::Element< TVR, TVM >::Internal](#)[[VMToLength< TVM >::Length](#)]

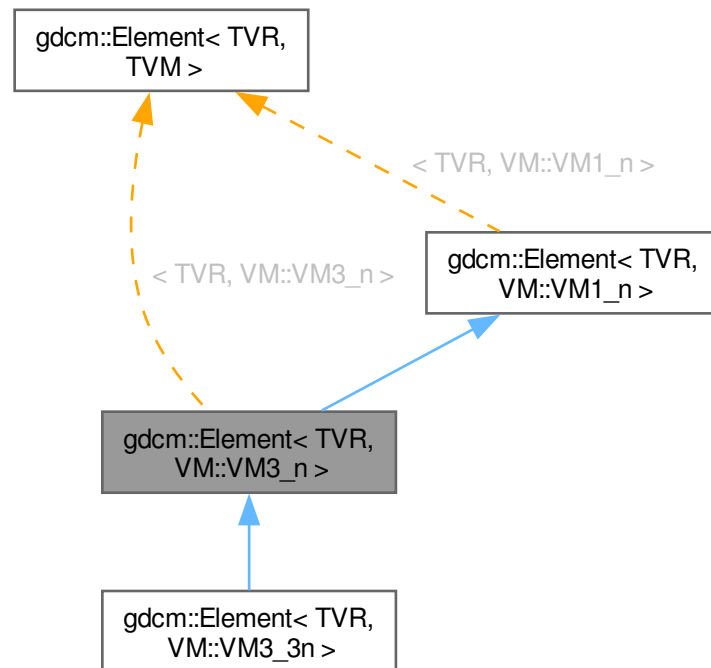
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

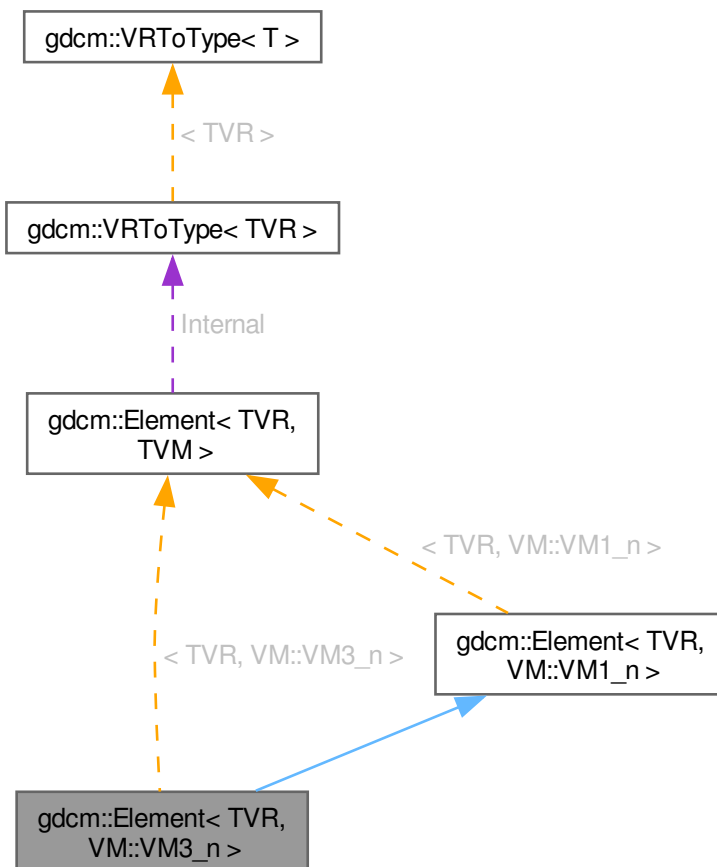
12.104 gdcmm::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcM::Element< TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_n >:



Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`
- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcm::Element< TVR, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`

- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (int len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- [Element](#) ()
- [Element](#) (const Element &_val)
- [~Element](#) ()
- [DataElement](#) [GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0)
- const [VRToType](#)< TVR >::Type & [GetValue](#) (unsigned int idx=0) const
- const [VRToType](#)< TVR >::Type * [GetValues](#) () const
- [Element](#) & operator= (const [Element](#) &_val)
- [VRToType](#)< TVR >::Type operator[] (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) (Value const &v)
- void [SetArray](#) (const Type *array, unsigned long len, bool save=false)
- void [SetFromDataElement](#) (DataElement const &de)
- void [SetLength](#) (unsigned long len)
- void [SetValue](#) (typename [VRToType](#)< TVR >::Type v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const
- void [WriteASCII](#) (std::ostream &os) const

Static Public Member Functions

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element](#)< TVR, VM::VM1_n >

- static [VM](#) [GetVM](#) ()
- static [VR](#) [GetVR](#) ()

Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcM::Element< TVR, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.104.1 Member Typedef Documentation

12.104.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1\_n> gdcM::Element< TVR, VM::VM3\_n>::Parent
```

12.104.1.2 Type

```
typedef VRToType<TVR>::Type gdcM::Element< TVR, TVM >::Type
```

12.104.2 Member Function Documentation

12.104.2.1 GetAsDataElement()

```
DataElement gdcM::Element< TVR, TVM >::GetAsDataElement () const [inline]
```

12.104.2.2 GetLength()

```
unsigned long gdcM::Element< TVR, TVM >::GetLength () const [inline]
```

12.104.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcM::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0) const [inline]
```

12.104.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcM::Element< TVR, TVM >::GetValues () const [inline]
```

12.104.2.5 GetVM()

[VM](#) [gdcm::Element](#)< TVR, TVM >::GetVM () [inline], [static]

12.104.2.6 GetVR()

[VR](#) [gdcm::Element](#)< TVR, TVM >::GetVR () [inline], [static]

12.104.2.7 operator[]()

[VRToType](#)< TVR >::Type [gdcm::Element](#)< TVR, TVM >::operator[] (unsigned int idx) const [inline]

12.104.2.8 Print()

void [gdcm::Element](#)< TVR, TVM >::Print (std::ostream & _os) const [inline]

12.104.2.9 Read()

void [gdcm::Element](#)< TVR, TVM >::Read (std::istream & _is) [inline]

12.104.2.10 Set()

void [gdcm::Element](#)< TVR, TVM >::Set ([Value](#) const & v) [inline]

12.104.2.11 SetFromDataElement()

void [gdcm::Element](#)< TVR, TVM >::SetFromDataElement ([DataElement](#)< TVR, [VM::VM3_n](#) > const & de) [inline]

12.104.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3\_n >::SetLength (
    int len) [inline]
```

References [gdcm::Element](#)< TVR, [VM::VM1_n](#) >::SetLength().

Referenced by [gdcm::Element](#)< TVR, [VM::VM3_3n](#) >::SetLength().

12.104.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

12.104.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```

12.104.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
    std::ostream & __os) const [inline]
```

12.104.3 Member Data Documentation

12.104.3.1 Internal

```
VRTToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

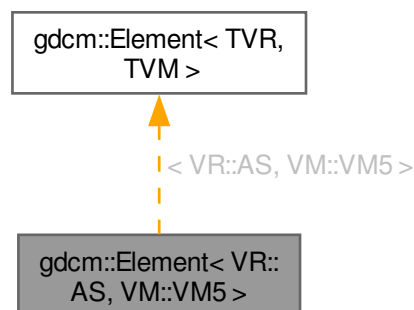
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

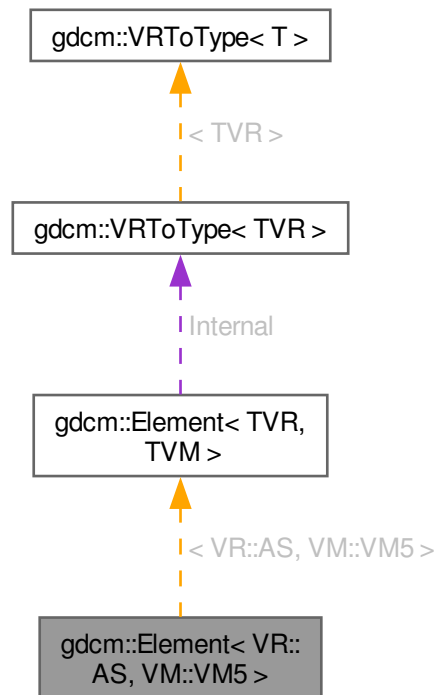
12.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::AS, VM::VM5 >:



Collaboration diagram for `gdcm::Element< VR::AS, VM::VM5 >`:



Public Types

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`
- `const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const`
- `const VRToType< TVR >::Type * GetValues () const`
- `VRToType< TVR >::Type operator[] (unsigned int idx) const`
- `void Print (std::ostream &_os) const`
- `void Read (std::istream &_is)`
- `void Set (Value const &v)`
- `void SetFromDataElement (DataElement const &de)`
- `void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)`
- `void Write (std::ostream &_os) const`

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- char [Internal](#) [[VMToLength](#)< [VM::VM5](#) >::Length *sizeof([VRToType](#)< [VR::AS](#) >::Type)]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

12.105.1 Member Typedef Documentation

12.105.1.1 Type

typedef [VRToType](#)<TVR>::Type [gdcm::Element](#)< TVR, TVM >::Type

12.105.2 Member Function Documentation

12.105.2.1 GetAsDataElement()

[DataElement](#) [gdcm::Element](#)< TVR, TVM >::GetAsDataElement () const [inline]

12.105.2.2 GetLength()

unsigned long [gdcm::Element](#)< [VR::AS](#), [VM::VM5](#) >::GetLength () const [inline]

12.105.2.3 GetValue()

const [VRToType](#)< TVR >::Type & [gdcm::Element](#)< TVR, TVM >::GetValue (unsigned int idx = 0) const [inline]

12.105.2.4 GetValues()

const [VRToType](#)< TVR >::Type * [gdcm::Element](#)< TVR, TVM >::GetValues () const [inline]

12.105.2.5 GetVM()

[VM](#) [gdcm::Element](#)< TVR, TVM >::GetVM () [inline], [static]

12.105.2.6 GetVR()

```
VR gdcmm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.105.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const [inline]
```

12.105.2.8 Print()

```
void gdcmm::Element< VR::AS, VM::VM5 >::Print (
    std::ostream & _os) const [inline]
```

References [Internal](#).

12.105.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & _is) [inline]
```

12.105.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v) [inline]
```

12.105.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< VR::AS, VM::VM5 > const & de) [inline]
```

12.105.2.12 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v) [inline], [protected]
```

12.105.2.13 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0) [inline]
```


12.105.2.14 Write()

```
void gdcm::Element< TVR, TVM >::Write (
    std::ostream & _os) const    [inline]
```

12.105.3 Member Data Documentation

12.105.3.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]
```

Referenced by [Print\(\)](#).

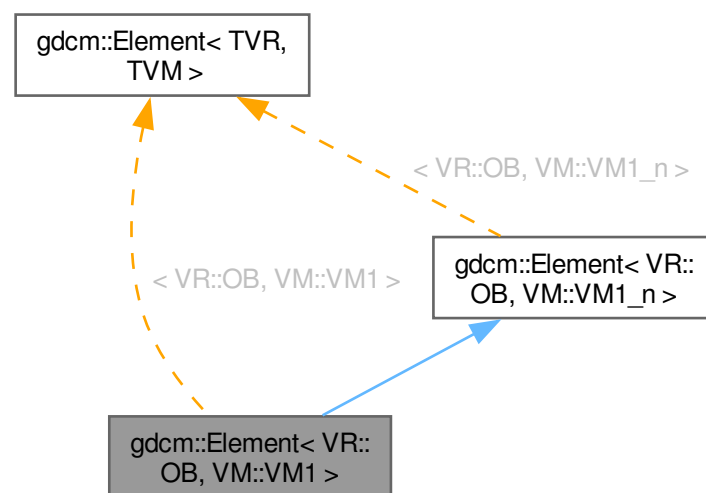
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

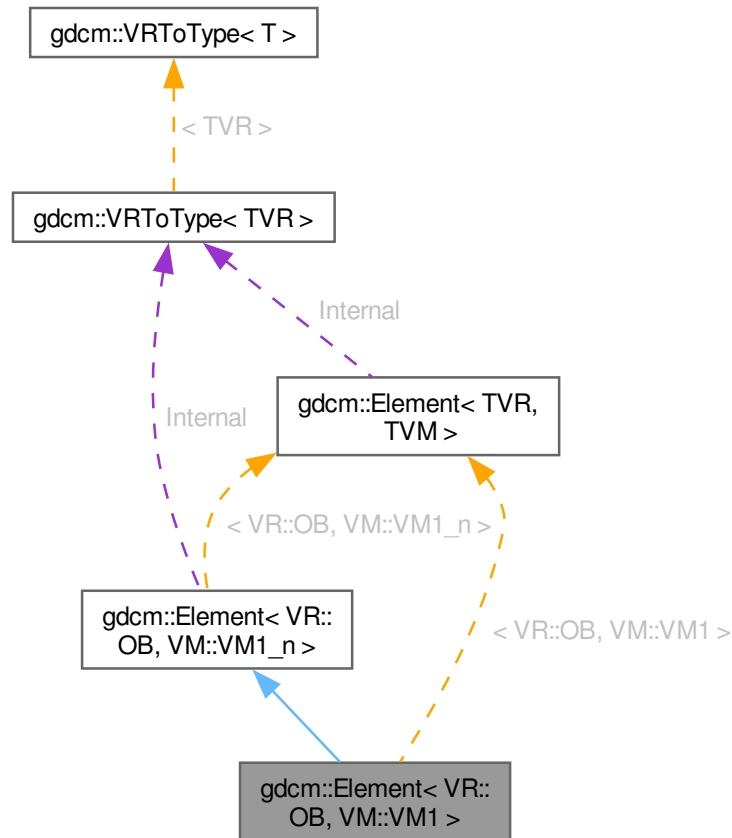
12.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for `gdcM::Element< VR::OB, VM::VM1 >`:



Public Types

- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcM::Element< VR::OB, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`
- `const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const`
- `const VRToType< TVR >::Type * GetValues () const`

- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element< VR::OB, VM::VM1_n >](#)

- [DataElement GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- const [VRToType< TVR >::Type](#) * [GetValues](#) () const
- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element< VR::OB, VM::VM1_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Public Attributes inherited from [gdcm::Element< VR::OB, VM::VM1_n >](#)

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcM::Element< VR::OB, VM::VM1_n >](#)

- void [SetNoSwap](#) (Value const &v)

12.106.1 Member Typedef Documentation

12.106.1.1 Type

typedef [VRToType](#)<TVR>::Type [gdcM::Element](#)< TVR, TVM >::Type

12.106.2 Member Function Documentation

12.106.2.1 GetAsDataElement()

[DataElement](#) [gdcM::Element](#)< TVR, TVM >::GetAsDataElement () const [inline]

12.106.2.2 GetLength()

unsigned long [gdcM::Element](#)< TVR, TVM >::GetLength () const [inline]

12.106.2.3 GetValue()

const [VRToType](#)< TVR >::Type & [gdcM::Element](#)< TVR, TVM >::GetValue (unsigned int idx = 0) const [inline]

12.106.2.4 GetValues()

const [VRToType](#)< TVR >::Type * [gdcM::Element](#)< TVR, TVM >::GetValues () const [inline]

12.106.2.5 GetVM()

[VM](#) [gdcM::Element](#)< TVR, TVM >::GetVM () [inline], [static]

12.106.2.6 GetVR()

[VR](#) [gdcM::Element](#)< TVR, TVM >::GetVR () [inline], [static]

12.106.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const    [inline]
```

12.106.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
    std::ostream & __os) const    [inline]
```

12.106.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
    std::istream & __is)    [inline]
```

12.106.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
    Value const & v)    [inline]
```

12.106.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< VR::OB, VM::VM1 > const & de)    [inline]
```

12.106.2.12 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
    Value const & v)    [inline], [protected]
```

12.106.2.13 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0)    [inline]
```

12.106.2.14 Write()

```
void gdcm::Element< TVR, TVM >::Write (
    std::ostream & __os) const    [inline]
```

12.106.3 Member Data Documentation

12.106.3.1 Internal

[VRToType<TVR>::Type](#) [gdcm::Element< TVR, TVM >::Internal](#)[[VMToLength< TVM >::Length](#)]

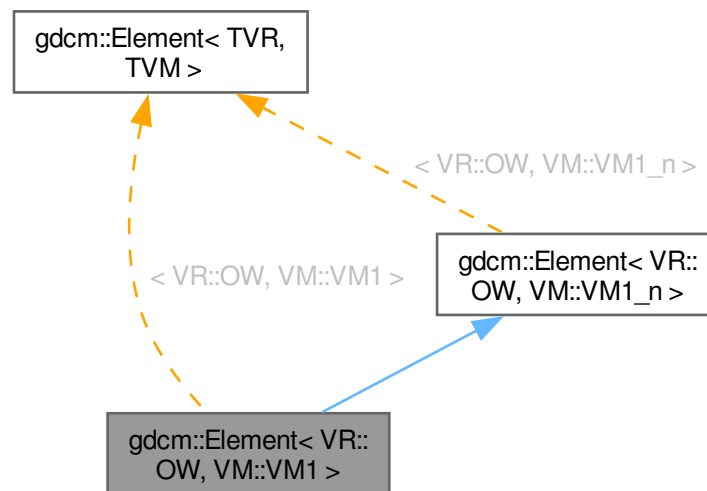
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

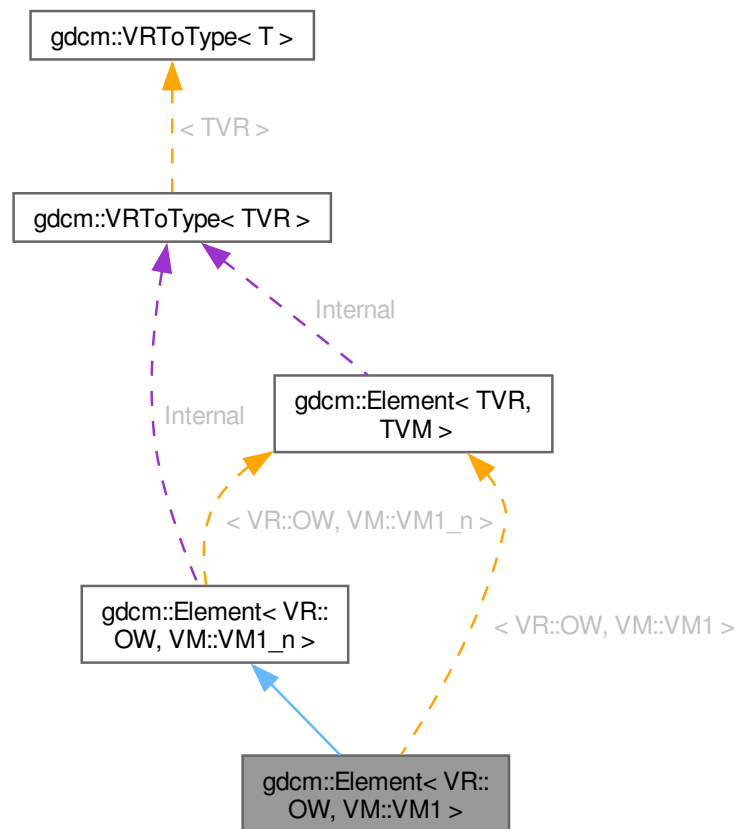
12.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< VR::OW, VM::VM1 >`:



Collaboration diagram for gdcM::Element< VR::OW, VM::VM1 >:



Public Types

- typedef `VRToType< TVR >::Type` `Type`

Public Types inherited from `gdcM::Element< VR::OW, VM::VM1_n >`

- typedef `VRToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement () const`
- `unsigned long GetLength () const`
- `const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const`
- `const VRToType< TVR >::Type * GetValues () const`

- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Public Member Functions inherited from [gdcm::Element< VR::OW, VM::VM1_n >](#)

- [DataElement GetAsDataElement](#) () const
- unsigned long [GetLength](#) () const
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- const [VRToType< TVR >::Type](#) * [GetValues](#) () const
- [VRToType< TVR >::Type operator\[\]](#) (unsigned int idx) const
- void [Print](#) (std::ostream &_os) const
- void [Read](#) (std::istream &_is)
- void [Set](#) ([Value](#) const &v)
- void [SetFromDataElement](#) ([DataElement](#) const &de)
- void [SetValue](#) (typename [VRToType< TVR >::Type](#) v, unsigned int idx=0)
- void [Write](#) (std::ostream &_os) const

Static Public Member Functions

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Static Public Member Functions inherited from [gdcm::Element< VR::OW, VM::VM1_n >](#)

- static [VM GetVM](#) ()
- static [VR GetVR](#) ()

Public Attributes

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Public Attributes inherited from [gdcm::Element< VR::OW, VM::VM1_n >](#)

- [VRToType< TVR >::Type Internal](#) [[VMToLength< TVM >::Length](#)]

Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

Protected Member Functions inherited from [gdcm::Element< VR::OW, VM::VM1_n >](#)

- void [SetNoSwap](#) ([Value](#) const &v)

12.107.1 Member Typedef Documentation

12.107.1.1 Type

typedef [VRToType](#)<TVR>::Type [gdcm::Element](#)< TVR, TVM >::Type

12.107.2 Member Function Documentation

12.107.2.1 GetAsDataElement()

[DataElement](#) [gdcm::Element](#)< TVR, TVM >::GetAsDataElement () const [inline]

12.107.2.2 GetLength()

unsigned long [gdcm::Element](#)< TVR, TVM >::GetLength () const [inline]

12.107.2.3 GetValue()

const [VRToType](#)< TVR >::Type & [gdcm::Element](#)< TVR, TVM >::GetValue (
unsigned int idx = 0) const [inline]

12.107.2.4 GetValues()

const [VRToType](#)< TVR >::Type * [gdcm::Element](#)< TVR, TVM >::GetValues () const [inline]

12.107.2.5 GetVM()

[VM](#) [gdcm::Element](#)< TVR, TVM >::GetVM () [inline], [static]

12.107.2.6 GetVR()

[VR](#) [gdcm::Element](#)< TVR, TVM >::GetVR () [inline], [static]

12.107.2.7 operator[]()

```
VRToType< TVR >::Type gdcmm::Element< TVR, TVM >::operator[] (
    unsigned int idx) const    [inline]
```

12.107.2.8 Print()

```
void gdcmm::Element< TVR, TVM >::Print (
    std::ostream & __os) const    [inline]
```

12.107.2.9 Read()

```
void gdcmm::Element< TVR, TVM >::Read (
    std::istream & __is)    [inline]
```

12.107.2.10 Set()

```
void gdcmm::Element< TVR, TVM >::Set (
    Value const & v)    [inline]
```

12.107.2.11 SetFromDataElement()

```
void gdcmm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< VR::OW, VM::VM1 > const & de)    [inline]
```

12.107.2.12 SetNoSwap()

```
void gdcmm::Element< TVR, TVM >::SetNoSwap (
    Value const & v)    [inline], [protected]
```

12.107.2.13 SetValue()

```
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0)    [inline]
```

12.107.2.14 Write()

```
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & __os) const    [inline]
```

12.107.3 Member Data Documentation

12.107.3.1 Internal

[VRToType](#)<TVR>::Type [gdcm::Element](#)< TVR, TVM >::Internal[[VMToLength](#)< TVM >::Length]

The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

12.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference

A class which is used to produce compile errors for an invalid combination of template parameters.

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::ElementDisableCombinations< TVR, TVM >`:



12.108.1 Detailed Description

```
template<long long TVR, int TVM>
class gdcm::ElementDisableCombinations< TVR, TVM >
```

A class which is used to produce compile errors for an invalid combination of template parameters.

Invalid combinations have specialized declarations with no definition.

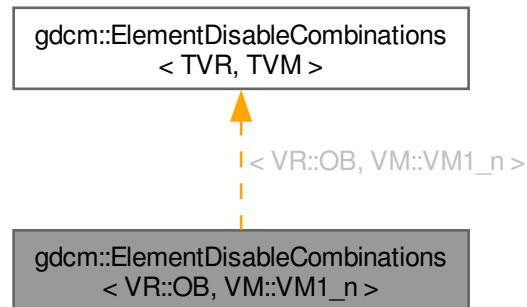
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

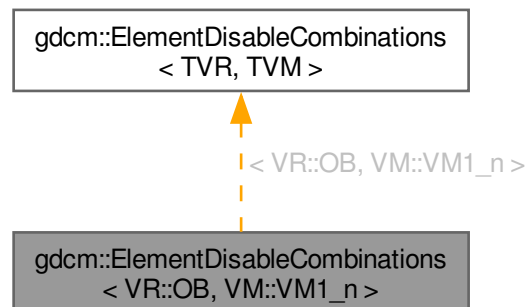
12.109 gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference

#include <gdcmmElement.h>

Inheritance diagram for gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >:



Collaboration diagram for gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >:



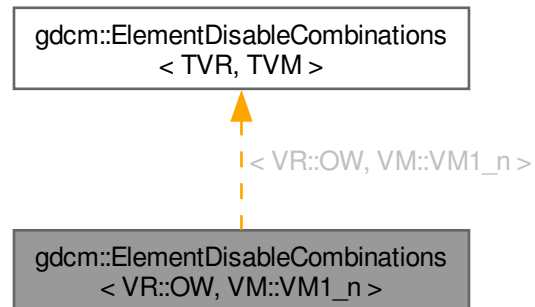
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

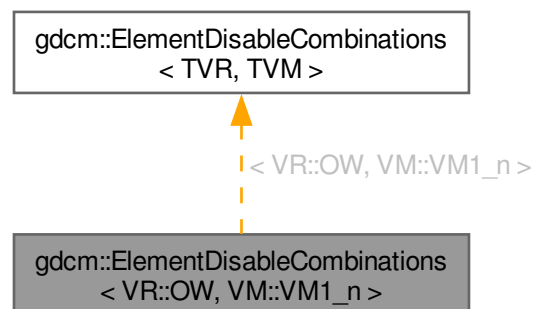
12.110 gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference

#include <gdcmmElement.h>

Inheritance diagram for gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >:



Collaboration diagram for gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >:



The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

12.111 gdcm::EmptyMaskGenerator Class Reference

[EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.

```
#include <gdcmEmptyMaskGenerator.h>
```

Public Types

- enum [SOPClassUIDMode](#) {
[UseOriginalSOPClassUID](#) = 0 ,
[UseGrayscaleSecondaryImageStorage](#) }

Public Member Functions

- [EmptyMaskGenerator](#) ()
- [~EmptyMaskGenerator](#) ()
- bool [Execute](#) ()
Main loop.
- void [SetInputDirectory](#) (const char *dirname)
Specify input directory.
- void [SetOutputDirectory](#) (const char *dirname)
Specify output directory.
- void [SetSOPClassUIDMode](#) ([SOPClassUIDMode](#) mode)

12.111.1 Detailed Description

[EmptyMaskGenerator](#) Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.

The class allow two mode of operations:

- [UseOriginalSOPClassUID](#)
- [UseGrayscaleSecondaryImageStorage](#)

[UseOriginalSOPClassUID](#) is the mode where original attributes are copied from the original DICOM instance.

[UseGrayscaleSecondaryImageStorage](#) is the mode where attributes are generated so as to create a [MultiframeGrayscaleByteSecondaryCaptureImageStorage](#) (MultiframeGrayscaleWordSecondaryCaptureImageStorage) instance.

In both mode:

- the [Study](#) references (StudyInstanceUID and StudyID) are preserved.

- the PatientID reference is preserved.
- the [Image Type](#) attribute will be setup so that the fourth element is set to 'MASK'.
- a new [Series](#) Instance UID is generated. It is thus required to run the process over all files using the same input [Series](#) Instance UID so that a proper mapping from the old [Series](#) UID is done to the new one. Since a new [Series](#) Instance UID is generated, there is no sense to preserve the original Frame of Reference UID, although it would have made sense here.

Examples

[EmptyMask.cxx](#).

12.111.2 Member Enumeration Documentation

12.111.2.1 SOPClassUIDMode

enum [gdcm::EmptyMaskGenerator::SOPClassUIDMode](#)

Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

12.111.3 Constructor & Destructor Documentation

12.111.3.1 EmptyMaskGenerator()

gdcm::EmptyMaskGenerator::EmptyMaskGenerator ()

12.111.3.2 ~EmptyMaskGenerator()

gdcm::EmptyMaskGenerator::~~EmptyMaskGenerator ()

12.111.4 Member Function Documentation

12.111.4.1 Execute()

bool gdcm::EmptyMaskGenerator::Execute ()

Main loop.

Examples

[EmptyMask.cxx](#).

12.111.4.2 SetInputDirectory()

```
void gdcmm::EmptyMaskGenerator::SetInputDirectory (  
    const char * dirname)
```

Specify input directory.

Examples

[EmptyMask.cxx](#).

12.111.4.3 SetOutputDirectory()

```
void gdcmm::EmptyMaskGenerator::SetOutputDirectory (  
    const char * dirname)
```

Specify output directory.

Examples

[EmptyMask.cxx](#).

12.111.4.4 SetSOPClassUIDMode()

```
void gdcmm::EmptyMaskGenerator::SetSOPClassUIDMode (  
    SOPClassUIDMode mode)
```

Select generation of SOP Class UID method: Default is UseOriginalSOPClassUID

Examples

[EmptyMask.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmmEmptyMaskGenerator.h](#)

12.112 gdcmm::EncapsulatedDocument Class Reference

[EncapsulatedDocument](#).

```
#include <gdcmmEncapsulatedDocument.h>
```


Public Member Functions

- [EncapsulatedDocument](#) ()=default

12.112.1 Detailed Description

[EncapsulatedDocument](#).

12.112.2 Constructor & Destructor Documentation

12.112.2.1 EncapsulatedDocument()

gdcm::EncapsulatedDocument::EncapsulatedDocument () [default]

The documentation for this class was generated from the following file:

- [gdcmEncapsulatedDocument.h](#)

12.113 gdcm::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

Inheritance diagram for gdcm::EncodingImplementation< T >:



12.113.1 Detailed Description

```
template<long long T>
class gdcm::EncodingImplementation< T >
```

[EncodingImplementation](#).

Note

TODO

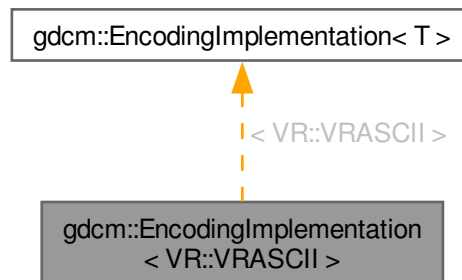
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

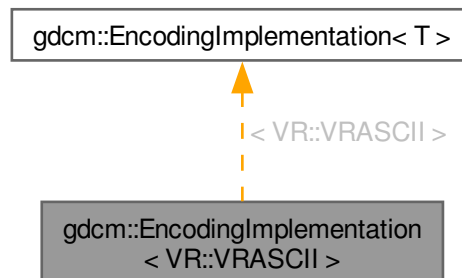
12.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::EncodingImplementation< VR::VRASCII >:



Collaboration diagram for gdcm::EncodingImplementation< VR::VRASCII >:



Public Member Functions

- template<> void [Write](#) (const double *data, unsigned long length, std::ostream &_os)

Static Public Member Functions

- `template<typename T>`
static void [Read](#) (T *data, unsigned long length, std::istream &_is)
- `template<typename T>`
static void [ReadComputeLength](#) (T *data, unsigned int &length, std::istream &_is)
- `template<typename T>`
static void [ReadNoSwap](#) (T *data, unsigned long length, std::istream &_is)
- `template<typename T>`
static void [Write](#) (const T *data, unsigned long length, std::ostream &_os)

12.114.1 Member Function Documentation

12.114.1.1 [Read\(\)](#)

```
template<typename T>
void gdcmm::EncodingImplementation< VR::VRASCII >::Read (
    T * data,
    unsigned long length,
    std::istream & _is) [inline], [static]
```

References [gdcmm_assert](#).

Referenced by [ReadNoSwap\(\)](#).

12.114.1.2 [ReadComputeLength\(\)](#)

```
template<typename T>
void gdcmm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is) [inline], [static]
```

References [gdcmm::backslash\(\)](#), and [gdcmm_assert](#).

12.114.1.3 [ReadNoSwap\(\)](#)

```
template<typename T>
void gdcmm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is) [inline], [static]
```

References [Read\(\)](#).

12.114.1.4 Write() [1/2]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const double * data,
    unsigned long length,
    std::ostream & _os) [inline]
```

References [gdcm_assert](#), and [gdcm::x16printf\(\)](#).

12.114.1.5 Write() [2/2]

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const T * data,
    unsigned long length,
    std::ostream & _os) [inline], [static]
```

References [gdcm_assert](#).

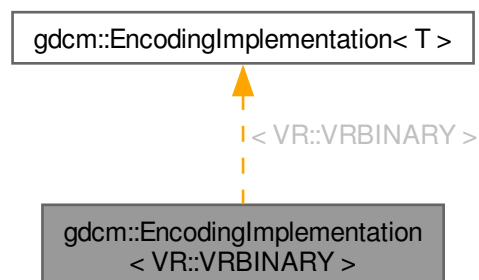
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

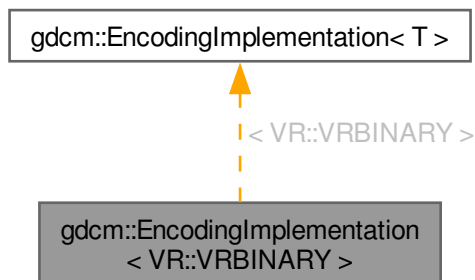
12.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::EncodingImplementation< VR::VRBINARY >:



Collaboration diagram for gdcm::EncodingImplementation< VR::VRBINARY >:



Static Public Member Functions

- `template<typename T>`
static void [Read](#) (T *data, unsigned long length, std::istream &_is)
- `template<typename T>`
static void [ReadComputeLength](#) (T *data, unsigned int &length, std::istream &_is)
- `template<typename T>`
static void [ReadNoSwap](#) (T *data, unsigned long length, std::istream &_is)
- `template<typename T>`
static void [Write](#) (const T *data, unsigned long length, std::ostream &_os)

12.115.1 Member Function Documentation

12.115.1.1 Read()

```

template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
    T * data,
    unsigned long length,
    std::istream & _is)  [inline], [static]
  
```

References [gdcm_assert](#), and [gdcm::SwapperNoOp::SwapArray\(\)](#).

12.115.1.2 ReadComputeLength()

```

template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is)  [inline], [static]
  
```

References [gdcm_assert](#).

12.115.1.3 ReadNoSwap()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & __is) [inline], [static]
```

References [gdcm__assert](#).

12.115.1.4 Write()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
    const T * data,
    unsigned long length,
    std::ostream & __os) [inline], [static]
```

References [gdcm__assert](#), and [gdcm::SwapperNoOp::Swap\(\)](#).

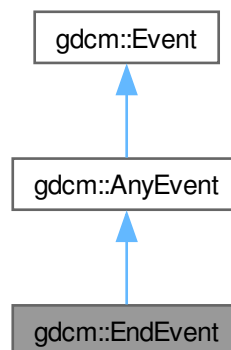
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

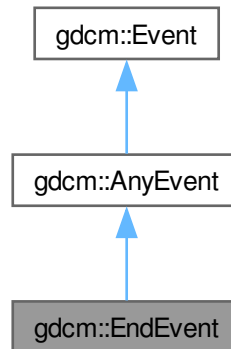
12.116 gdcm::EndEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for `gdcm::EndEvent`:



Collaboration diagram for gdcM::EndEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcM::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcMEvent.h](#)

12.117 gdcM::EnumeratedValues Class Reference

Element. A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

```
#include <gdcMEnumeratedValues.h>
```

Public Member Functions

- [EnumeratedValues](#) ()=default

12.117.1 Detailed Description

Element. A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

1. [Patient](#) Sex (0010, 0040) is an example of a Data [Element](#) having Enumerated Values. It is defined to have a [Value](#) that is either "M", "F", or "O" (see PS 3.3). No other [Value](#) shall be given to this Data [Element](#).
2. Future modifications of this standard may add to the set of allowed values for Data Elements with Enumerated Values. Such additions by themselves may or may not require a change in SOP Class [UIDs](#), depending on the semantics of the Data [Element](#).

12.117.2 Constructor & Destructor Documentation

12.117.2.1 EnumeratedValues()

gdcm::EnumeratedValues::EnumeratedValues () [default]

The documentation for this class was generated from the following file:

- [gdcmEnumeratedValues.h](#)

12.118 gdcm::EquipmentManufacturer Class Reference

```
#include <gdcmEquipmentManufacturer.h>
```

Public Types

- enum [Type](#) {
[UNKNOWN](#) = 0 ,
[AGFA](#) ,
[FUJI](#) ,
[GEMS](#) ,
[HITACHI](#) ,
[KODAK](#) ,
[MARCONI](#) ,
[PMS](#) ,
[SAMSUNG](#) ,
[SIEMENS](#) ,
[TOSHIBA](#) ,
[UIH](#) }

Static Public Member Functions

- static [Type Compute](#) ([DataSet](#) const &ds)
- static const char * [TypeToString](#) ([Type](#) type)

12.118.1 Detailed Description

The intent is for private tags handling. This class is not meant to handle all possible vendors in the world, simply those well known where we intend to read private tags afterwards (typically SIEMENS+CSA, GEMS+↵ PDB ...)

12.118.2 Member Enumeration Documentation

12.118.2.1 Type

enum [gdcm::EquipmentManufacturer::Type](#)

Enumerator

UNKNOWN	
AGFA	
FUJI	
GEMS	
HITACHI	
KODAK	
MARCONI	
PMS	
SAMSUNG	
SIEMENS	
TOSHIBA	
UIH	

12.118.3 Member Function Documentation

12.118.3.1 Compute()

[Type](#) [gdcm::EquipmentManufacturer::Compute](#) (
 [DataSet](#) const & ds) [static]

12.118.3.2 TypeToString()

```
const char * gdcM::EquipmentManufacturer::TypeToString (  
    Type type) [static]
```

The documentation for this class was generated from the following file:

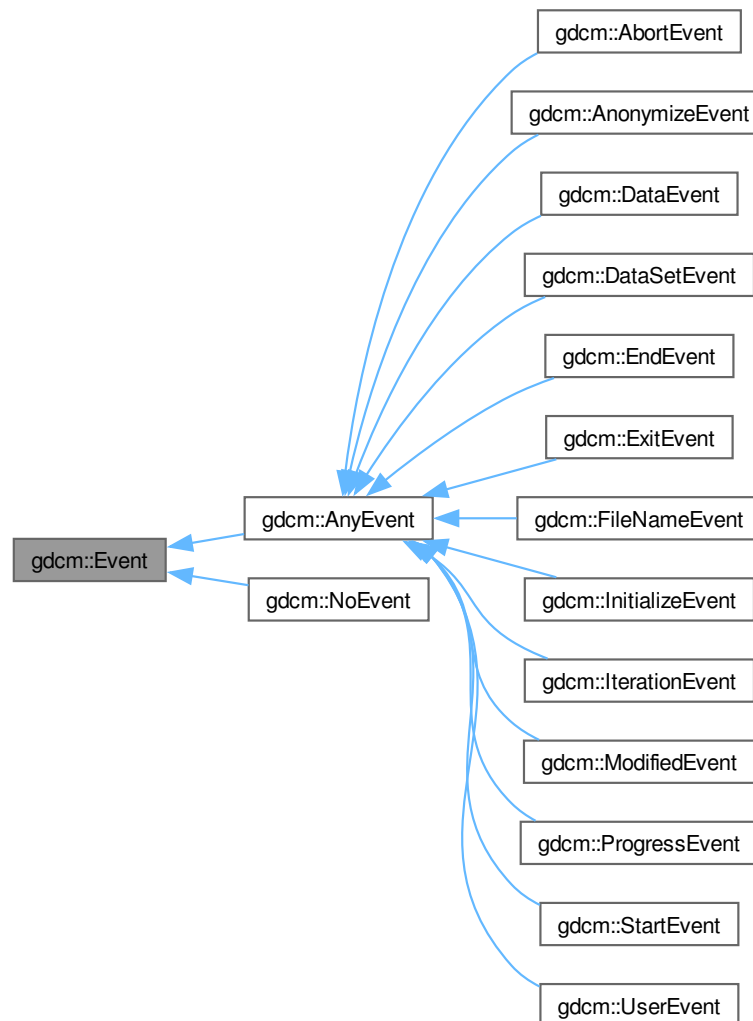
- [gdcMEquipmentManufacturer.h](#)

12.119 gdcM::Event Class Reference

superclass for callback/observer methods

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::Event:



Public Member Functions

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

12.119.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.119.2 Constructor & Destructor Documentation

12.119.2.1 [Event\(\)](#) [1/2]

gdcm::Event::Event ()

Referenced by [Event\(\)](#), [CheckEvent\(\)](#), [MakeObject\(\)](#), and [operator=\(\)](#).

12.119.2.2 [~Event\(\)](#)

virtual gdcm::Event::~~Event () [virtual]

12.119.2.3 [Event\(\)](#) [2/2]

gdcm::Event::Event (
 const [Event](#) &)

References [Event\(\)](#).

12.119.3 Member Function Documentation

12.119.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
    const Event * ) const    [pure virtual]
```

Check if given event matches or derives from this event.

References [Event\(\)](#).

12.119.3.2 GetEventName()

```
virtual const char * gdcm::Event::GetEventName () const    [pure virtual]
```

Return the StringName associated with the event.

Implemented in [gdcm::AnonymizeEvent](#), [gdcm::DataEvent](#), [gdcm::DataSetEvent](#), [gdcm::FileNameEvent](#), and [gdcm::ProgressEvent](#).

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), and [ScanDirectory.cs](#).

12.119.3.3 MakeObject()

```
virtual Event * gdcm::Event::MakeObject () const    [pure virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implemented in [gdcm::AnonymizeEvent](#), [gdcm::DataEvent](#), [gdcm::DataSetEvent](#), [gdcm::FileNameEvent](#), and [gdcm::ProgressEvent](#).

References [Event\(\)](#).

12.119.3.4 operator=()

```
void gdcm::Event::operator= (
    const Event & )    [delete]
```

References [Event\(\)](#).

12.119.3.5 Print()

```
virtual void gdcm::Event::Print (  
    std::ostream & os) const    [virtual]
```

Print [Event](#) information. This method can be overridden by specific [Event](#) subtypes. The default is to print out the type of the event.

Referenced by [gdcm::operator<<\(\)](#).

The documentation for this class was generated from the following file:

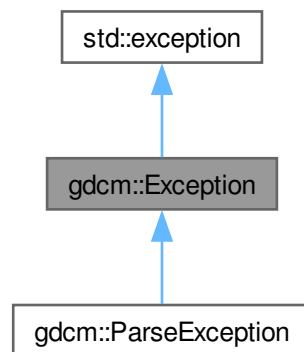
- [gdcmEvent.h](#)

12.120 gdcm::Exception Class Reference

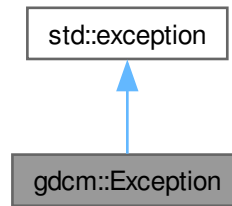
[Exception](#).

```
#include <gdcmException.h>
```

Inheritance diagram for gdcm::Exception:



Collaboration diagram for `gdcm::Exception`:



Public Member Functions

- [Exception](#) (`const char *desc="None", const char *file=__FILE__, unsigned int lineNumber=__LINE__, const char *func=""`)
- [~Exception](#) () override throw ()
- `const char *` [GetDescription](#) () const
Return the Description.
- `const char *` [what](#) () const override throw ()
what implementation

12.120.1 Detailed Description

[Exception](#).

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

12.120.2 Constructor & Destructor Documentation

12.120.2.1 Exception()

```
gdcmm::Exception::Exception (
    const char * desc = "None",
    const char * file = __FILE__,
    unsigned int lineNumber = __LINE__,
    const char * func = "") [inline], [explicit]
```

Explicit constructor, initializing the description and the text returned by [what\(\)](#).

Note

The last parameter is ignored for the time being. It may be used to specify the function where the exception was thrown.

Referenced by [gdcmm::ParseException::ParseException\(\)](#).

12.120.2.2 ~Exception()

```
gdcmm::Exception::~Exception () throw ( ) [inline], [override]
```

12.120.3 Member Function Documentation

12.120.3.1 GetDescription()

```
const char * gdcmm::Exception::GetDescription () const [inline]
```

Return the Description.

Referenced by [gdcmm::SequenceOfItems::Read\(\)](#).

12.120.3.2 what()

```
const char * gdcmm::Exception::what () const throw ( ) [inline], [override]
```

what implementation

Referenced by [gdcmm::SequenceOfFragments::ReadValue\(\)](#).

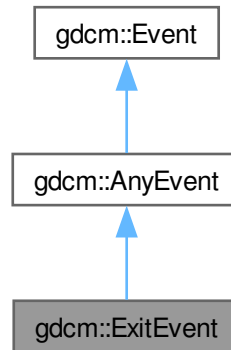
The documentation for this class was generated from the following file:

- [gdcmmException.h](#)

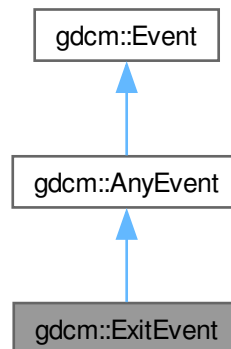
12.121 gdcM::ExitEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::ExitEvent:



Collaboration diagram for gdcM::ExitEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcM::Event](#)

- [Event](#) ()

- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

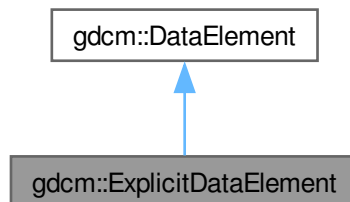
- [gdcmEvent.h](#)

12.122 gdcm::ExplicitDataElement Class Reference

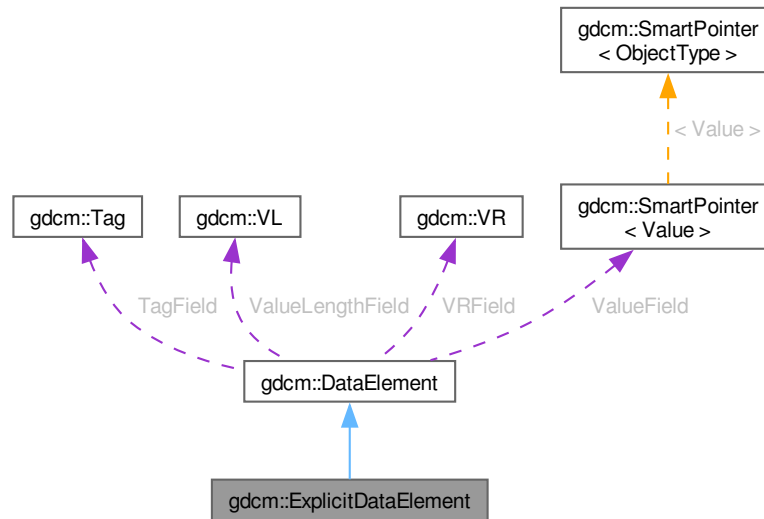
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#include <gdcmExplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitDataElement:



Collaboration diagram for `gdcm::ExplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- template<typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const

- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
- Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
- Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
- Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
- Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
- return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcmm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.122.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Note

bla

Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), and [ReadAndDumpDICOMDIR2.cxx](#).

12.122.2 Member Function Documentation

12.122.2.1 GetLength()

[VL](#) [gdcm::ExplicitDataElement::GetLength](#) () const

12.122.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::Read (
    std::istream & is)
```

12.122.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.122.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```

12.122.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length)
```

12.122.2.6 Write()

```
template<typename TSwap>
const std::ostream & gdcm::ExplicitDataElement::Write (
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

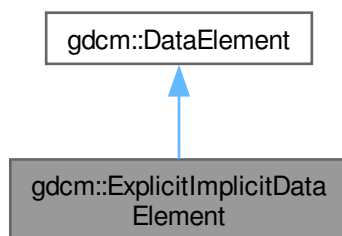
- [gdcmExplicitDataElement.h](#)

12.123 gdcm::ExplicitImplicitDataElement Class Reference

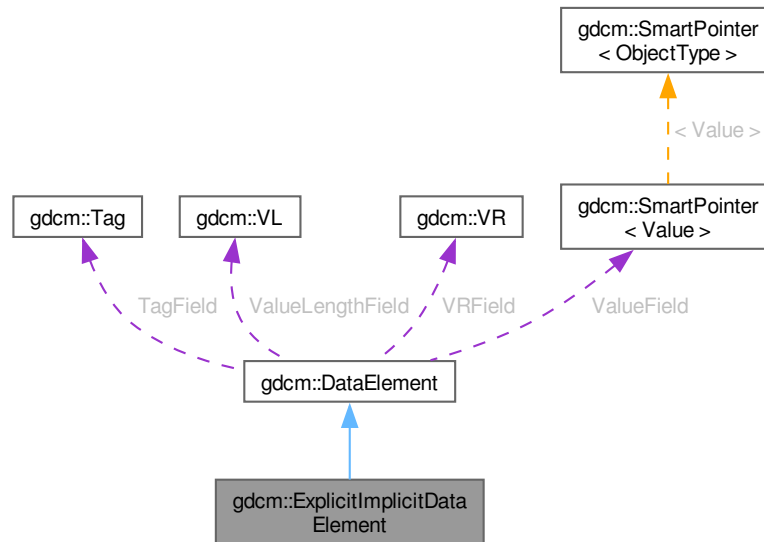
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitImplicitDataElement:



Collaboration diagram for `gdcm::ExplicitImplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()

- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
- [GetTag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
- Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
- [GetVL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
- Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
- return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.123.1 Detailed Description

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

Note

This only happen for some Philips images Should I derive from [ExplicitDataElement](#) instead ? This is the class that is the closest the GDCM1.x parser. At each element we try first to read it as explicit, if this fails, then we try again as an implicit element.

12.123.2 Member Function Documentation

12.123.2.1 GetLength()

[VL](#) [gdcm::ExplicitImplicitDataElement::GetLength \(\)](#) const

12.123.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::Read (
    std::istream & is)
```

Referenced by [ReadWithLength\(\)](#).

12.123.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.123.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```


12.123.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcM::ExplicitImplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length) [inline]
```

References [Read\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMExplicitImplicitDataElement.h](#)

12.124 gdcM::Fiducials Class Reference

[Fiducials](#).

```
#include <gdcMfiducials.h>
```

Public Member Functions

- [Fiducials](#) ()=default

12.124.1 Detailed Description

[Fiducials](#).

12.124.2 Constructor & Destructor Documentation

12.124.2.1 Fiducials()

```
gdcM::Fiducials::Fiducials () [default]
```

The documentation for this class was generated from the following file:

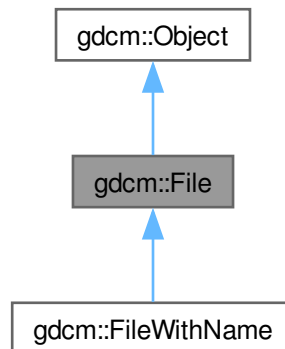
- [gdcMfiducials.h](#)

12.125 gdcM::File Class Reference

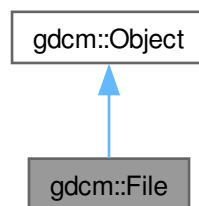
a DICOM [File](#)

```
#include <gdcmFile.h>
```

Inheritance diagram for gdcM::File:



Collaboration diagram for gdcM::File:



Public Member Functions

- [File](#) ()
- [~File](#) () override
- [DataSet](#) & [GetDataSet](#) ()
Get Data Set.

- const [DataSet](#) & [GetDataSet](#) () const
Get Data Set.
- [FileMetaInformation](#) & [GetHeader](#) ()
Get [File](#) Meta Information.
- const [FileMetaInformation](#) & [GetHeader](#) () const
Get [File](#) Meta Information.
- std::istream & [Read](#) (std::istream &is)
Read.
- void [SetDataSet](#) (const [DataSet](#) &ds)
Set Data Set.
- void [SetHeader](#) (const [FileMetaInformation](#) &fmi)
Set [File](#) Meta Information.
- std::ostream const & [Write](#) (std::ostream &os) const
Write.

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [File](#) &val)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.125.1 Detailed Description

a DICOM [File](#)

See PS 3.10 [File](#): A [File](#) is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the [File](#). Files are identified by a unique [File](#) ID and may be written, read and/or deleted.

See also

[Reader Writer](#)

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpCSA.cs](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [iU22tomultisc.cxx](#).

12.125.2 Constructor & Destructor Documentation

12.125.2.1 File()

`gdcm::File::File ()`

Referenced by [gdcm::FileWithName::FileWithName\(\)](#), [~File\(\)](#), and [operator<<](#).

12.125.2.2 ~File()

`gdcm::File::~File ()` [override]

References [File\(\)](#), and [operator<<](#).

12.125.3 Member Function Documentation

12.125.3.1 GetDataSet() [1/2]

[DataSet](#) & `gdcm::File::GetDataSet ()` [inline]

Get Data Set.

12.125.3.2 GetDataSet() [2/2]

const [DataSet](#) & gdcmm::File::GetDataSet () const [inline]

Get Data Set.

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrint.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.125.3.3 GetHeader() [1/2]

[FileMetaInformation](#) & gdcmm::File::GetHeader () [inline]

Get [File](#) Meta Information.

12.125.3.4 GetHeader() [2/2]

const [FileMetaInformation](#) & gdcmm::File::GetHeader () const [inline]

Get [File](#) Meta Information.

Examples

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [StreamImageReaderTest.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [operator<<](#).

12.125.3.5 Read()

std::istream & gdcmm::File::Read (
std::istream & is)

Read.

12.125.3.6 SetDataSet()

```
void gdcmm::File::SetDataSet (  
    const DataSet & ds) [inline]
```

Set Data Set.

12.125.3.7 SetHeader()

```
void gdcmm::File::SetHeader (  
    const FileMetaInformation & fmi) [inline]
```

Set File Meta Information.

12.125.3.8 Write()

```
std::ostream const & gdcmm::File::Write (  
    std::ostream & os) const
```

Write.

12.125.4 Friends And Related Symbol Documentation

12.125.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const File & val) [friend]
```

References [File\(\)](#), [gdcmm_assert](#), [GetHeader\(\)](#), and [operator<<](#).

Referenced by [~File\(\)](#), and [operator<<](#).

The documentation for this class was generated from the following file:

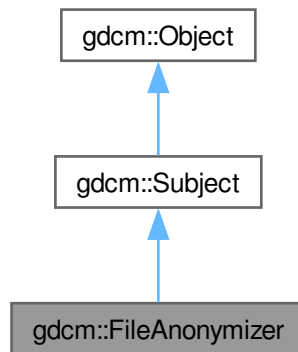
- [gdcmmFile.h](#)

12.126 gdcm::FileAnonymizer Class Reference

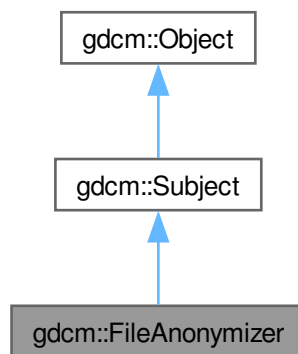
[FileAnonymizer](#).

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



Public Member Functions

- [FileAnonymizer](#) ()
- [~FileAnonymizer](#) () override
- void [Empty](#) ([Tag](#) const &t)
- void [Remove](#) ([Tag](#) const &t)
remove a tag (even a SQ can be removed)
- void [Replace](#) ([Tag](#) const &t, const char *value__data, [VL](#) const &vl)
- void [Replace](#) ([Tag](#) const &t, const char *value__str)
- void [SetInputFileName](#) (const char *filename__native)
Set input filename.
- void [SetOutputFileName](#) (const char *filename__native)
Set output filename.
- bool [Write](#) ()
Write the output file.

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.126.1 Detailed Description

[FileAnonymizer](#).

This [Anonymizer](#) is a file-based [Anonymizer](#). It requires a valid DICOM file and will use the [Value](#) Length to skip over any information.

It will not load the DICOM dataset taken from [SetInputFileName\(\)](#) into memory and should consume much less memory than [Anonymizer](#).

Warning

: Each time you call [Replace\(\)](#) with a value. This value will copied, and stored in memory. The behavior is not ideal for extremely large data (larger than memory size). This class is really meant to take a large DICOM input file and then only changed some small attribute.

caveats:

- This class will NOT work with unordered attributes in a DICOM [File](#),
- This class does neither recompute nor update the Group Length element,
- This class currently does not update the [File](#) Meta Information header.
- Only strict inplace Replace operation is supported when input and output file are the same.

Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

12.126.2 Constructor & Destructor Documentation

12.126.2.1 FileAnonymizer()

```
gdcm::FileAnonymizer::FileAnonymizer ()
```

12.126.2.2 ~FileAnonymizer()

```
gdcm::FileAnonymizer::~~FileAnonymizer () [override]
```

12.126.3 Member Function Documentation

12.126.3.1 Empty()

```
void gdcm::FileAnonymizer::Empty (
    Tag const & t)
```

Make [Tag](#) t empty Warning: does not handle SQ element

Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

12.126.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (  
    Tag const & t)
```

remove a tag (even a SQ can be removed)

Examples

[FileAnonymize.cs](#).

12.126.3.3 Replace() [1/2]

```
void gdcm::FileAnonymizer::Replace (  
    Tag const & t,  
    const char * value_data,  
    VL const & vl)
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

12.126.3.4 Replace() [2/2]

```
void gdcm::FileAnonymizer::Replace (  
    Tag const & t,  
    const char * value_str)
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII WARNING: Do not ever try to write a value in a SQ Data [Element](#) !

Examples

[FileAnonymize.cs](#).

12.126.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (  
    const char * filename_native)
```

Set input filename.

Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

12.126.3.6 SetOutputFileName()

```
void gdcm::FileAnonymizer::SetOutputFileName (  
    const char * filename__native)
```

Set output filename.

Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

12.126.3.7 Write()

```
bool gdcm::FileAnonymizer::Write ()
```

Write the output file.

Examples

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

The documentation for this class was generated from the following file:

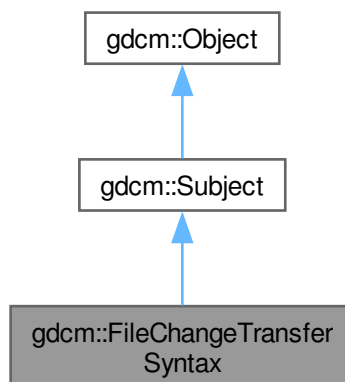
- [gdcmFileAnonymizer.h](#)

12.127 gdcm::FileChangeTransferSyntax Class Reference

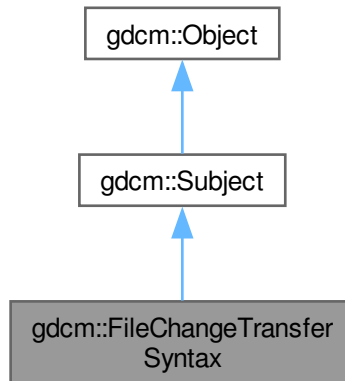
[FileChangeTransferSyntax](#).

```
#include <gdcmFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for `gdcm::FileChangeTransferSyntax`:



Public Member Functions

- [FileChangeTransferSyntax](#) ()
- [~FileChangeTransferSyntax](#) () override
- bool [Change](#) ()
Change the transfer syntax.
- [ImageCodec](#) * [GetCodec](#) ()
- void [SetInputFileName](#) (const char *filename__native)
Set input filename (raw DICOM).
- void [SetOutputFileName](#) (const char *filename__native)
Set output filename (target compressed DICOM).
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)
Specify the Target Transfer Syntax.

Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [FileChangeTransferSyntax](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.127.1 Detailed Description

[FileChangeTransferSyntax](#).

This class is a file-based (limited) replacement of the in-memory [ImageChangeTransferSyntax](#).

This class provide a file-based compression-only mechanism. It will take in an uncompressed DICOM image file (Pixel Data element). Then produced as output a compressed DICOM file (Transfer Syntax will be updated).

Currently it supports the following transfer syntax:

- JPEGLosslessProcess14_1

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

12.127.2 Constructor & Destructor Documentation

12.127.2.1 [FileChangeTransferSyntax](#)()

`gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ()`

Referenced by [New\(\)](#).

12.127.2.2 ~FileChangeTransferSyntax()

`gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax ()` [override]

12.127.3 Member Function Documentation

12.127.3.1 Change()

`bool gdcm::FileChangeTransferSyntax::Change ()`

Change the transfer syntax.

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

12.127.3.2 GetCodec()

`ImageCodec * gdcm::FileChangeTransferSyntax::GetCodec ()`

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

Examples

[FileChangeTSLossy.cs](#).

12.127.3.3 New()

`SmartPointer< FileChangeTransferSyntax > gdcm::FileChangeTransferSyntax::New ()` [inline], [static]

for wrapped language: instantiate a reference counted object

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

References [FileChangeTransferSyntax\(\)](#).

12.127.3.4 SetInputFileName()

`void gdcm::FileChangeTransferSyntax::SetInputFileName (`
`const char * filename__native)`

Set input filename (raw DICOM).

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

12.127.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (
    const char * filename_native)
```

Set output filename (target compressed DICOM).

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

12.127.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (
    TransferSyntax const & ts)
```

Specify the Target Transfer Syntax.

Examples

[FileChangeTS.cs](#), and [FileChangeTSLossy.cs](#).

The documentation for this class was generated from the following file:

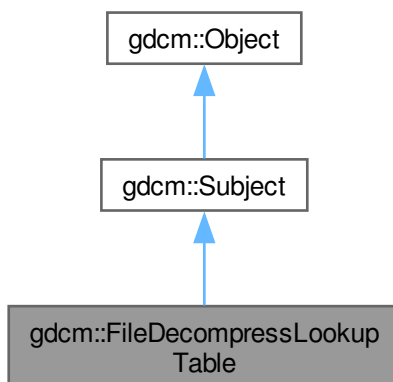
- [gdcmFileChangeTransferSyntax.h](#)

12.128 gdcm::FileDecompressLookupTable Class Reference

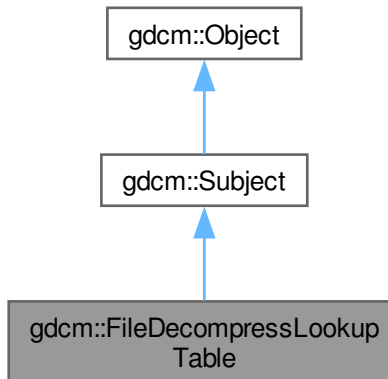
[FileDecompressLookupTable](#) class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for `gdcm::FileDecompressLookupTable`:



Public Member Functions

- `FileDecompressLookupTable ()`=default
- `~FileDecompressLookupTable ()` override=default
- `bool Change ()`
Decompress.
- `File & GetFile ()`
- `Pixmap & GetPixmap ()`
- `const Pixmap & GetPixmap () const`
- `void SetFile (const File &f)`
Set/Get `File`.
- `void SetPixmap (Pixmap const &img)`

Public Member Functions inherited from `gdcm::Subject`

- `Subject ()`
- `~Subject ()` override
- `unsigned long AddObserver (const Event &event, Command *)`
- `unsigned long AddObserver (const Event &event, Command *) const`
- `Command * GetCommand (unsigned long tag)`
- `bool HasObserver (const Event &event) const`
- `void InvokeEvent (const Event &)`
- `void InvokeEvent (const Event &) const`
- `void RemoveAllObservers ()`
- `void RemoveObserver (unsigned long tag)`

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.128.1 Detailed Description

[FileDecompressLookupTable](#) class.

It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image

12.128.2 Constructor & Destructor Documentation

12.128.2.1 [FileDecompressLookupTable](#)()

`gdcm::FileDecompressLookupTable::FileDecompressLookupTable ()` [default]

12.128.2.2 [~FileDecompressLookupTable](#)()

`gdcm::FileDecompressLookupTable::~~FileDecompressLookupTable ()` [override], [default]

12.128.3 Member Function Documentation

12.128.3.1 [Change](#)()

`bool gdcm::FileDecompressLookupTable::Change ()`

Decompress.

12.128.3.2 GetFile()

[File](#) & gdcm::FileDecompressLookupTable::GetFile () [inline]

12.128.3.3 GetPixmap() [1/2]

[Pixmap](#) & gdcm::FileDecompressLookupTable::GetPixmap () [inline]

12.128.3.4 GetPixmap() [2/2]

const [Pixmap](#) & gdcm::FileDecompressLookupTable::GetPixmap () const [inline]

12.128.3.5 SetFile()

void gdcm::FileDecompressLookupTable::SetFile (
 const [File](#) & f) [inline]

Set/Get [File](#).

12.128.3.6 SetPixmap()

void gdcm::FileDecompressLookupTable::SetPixmap (
 [Pixmap](#) const & img) [inline]

The documentation for this class was generated from the following file:

- [gdcmFileDecompressLookupTable.h](#)

12.129 gdcm::FileDerivation Class Reference

[FileDerivation](#) class.

```
#include <gdcmFileDerivation.h>
```

Public Member Functions

- [FileDerivation](#) ()
- [~FileDerivation](#) ()
- bool [AddReference](#) (const char *referencedsopclassuid, const char *referencedsopinstanceuid)
- bool [Derive](#) ()
Change.
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetAppendDerivationHistory](#) (bool b)
- void [SetDerivationCodeSequenceCodeValue](#) (unsigned int codevalue)
Specify the Derivation Code Sequence Code [Value](#). Eg 113040.
- void [SetDerivationDescription](#) (const char *dd)
Specify the Derivation Description. Eg "lossy conversion".
- void [SetFile](#) (const [File](#) &f)
Set/Get [File](#).
- void [SetPurposeOfReferenceCodeSequenceCodeValue](#) (unsigned int codevalue)
Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

Protected Member Functions

- bool [AddDerivationDescription](#) ()
- bool [AddPurposeOfReferenceCodeSequence](#) ([DataSet](#) &ds)
- bool [AddSourceImageSequence](#) ()

12.129.1 Detailed Description

[FileDerivation](#) class.

See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence

URL: http://medical.nema.org/medical/dicom/2008/08_16pu.pdf

DICOM Part 16 has two Context Groups CID 7202 and CID 7203 which contain a set of codes defining reason for a source image reference (ie. reason code for referenced image sequence) and a coded description of the deriation applied to the new image data from the original. Both these context groups are extensible.

[File](#) Derivation is compulsory when creating a lossy derived image.

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.2 Constructor & Destructor Documentation

12.129.2.1 FileDerivation()

gdcm::FileDerivation::FileDerivation ()

12.129.2.2 ~FileDerivation()

gdcm::FileDerivation::~~FileDerivation ()

12.129.3 Member Function Documentation

12.129.3.1 AddDerivationDescription()

bool gdcm::FileDerivation::AddDerivationDescription () [protected]

12.129.3.2 AddPurposeOfReferenceCodeSequence()

bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
 [DataSet](#) & ds) [protected]

12.129.3.3 AddReference()

bool gdcm::FileDerivation::AddReference (
 const char * referencedsopclassuid,
 const char * referencedsopinstanceuid)

Create the proper reference. Need to pass the original SOP Class UID and the original SOP Instance UID, so that those value can be used as Reference.

Warning

referencedsopclassuid and referencedsopinstanceuid needs to be \0 padded. This is not compatible with how `ByteValue->GetPointer` works.

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.3.4 AddSourceImageSequence()

bool gdcm::FileDerivation::AddSourceImageSequence () [protected]

12.129.3.5 Derive()

bool gdcm::FileDerivation::Derive ()

Change.

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.3.6 GetFile() [1/2]

[File](#) & gdcm::FileDerivation::GetFile () [inline]

Examples

[GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.3.7 GetFile() [2/2]

const [File](#) & gdcm::FileDerivation::GetFile () const [inline]

12.129.3.8 SetAppendDerivationHistory()

```
void gdcm::FileDerivation::SetAppendDerivationHistory (  
    bool b)
```

Specify if Derivation history should be appended (default false) When false, this is an error if input already has a derivation history When true, both Purpose of Reference Code [Value](#) and Derivation Code Sequence Code [Value](#) can have their history appended.

12.129.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (  
    unsigned int codevalue)
```

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.3.10 SetDerivationDescription()

```
void gdcm::FileDerivation::SetDerivationDescription (  
    const char * dd)
```

Specify the Derivation Description. Eg "lossy conversion".

12.129.3.11 SetFile()

```
void gdcmm::FileDerivation::SetFile (
    const File & f) [inline]
```

Set/Get [File](#).

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

12.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcmm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
    unsigned int codevalue)
```

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

Examples

[DeriveSeries.cxx](#), [GenFakeImage.cxx](#), and [ReformatFile.cs](#).

The documentation for this class was generated from the following file:

- [gdcmmFileDerivation.h](#)

12.130 gdcmm::FileExplicitFilter Class Reference

[FileExplicitFilter](#) class.

```
#include <gdcmmFileExplicitFilter.h>
```

Public Member Functions

- [FileExplicitFilter](#) ()
- [~FileExplicitFilter](#) ()=default
- bool [Change](#) ()
Set FMI Transfer Syntax.
- [File](#) & [GetFile](#) ()
- void [SetChangePrivateTags](#) (bool b)
Decide whether or not to [VR](#)'ify private tags.
- void [SetFile](#) (const [File](#) &f)
Set/Get [File](#).
- void [SetRecomputeItemLength](#) (bool b)
By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:
- void [SetRecomputeSequenceLength](#) (bool b)
- void [SetUseVRUN](#) (bool b)
When [VR](#)=16bits in explicit but Implicit has a 32bits length, use [VR](#)=UN.

Protected Member Functions

- bool [ChangeFMI](#) ()
- bool [ProcessDataSet](#) ([DataSet](#) &ds, [Dicts](#) const &dicts)

12.130.1 Detailed Description

[FileExplicitFilter](#) class.

After changing a file from Implicit to Explicit representation (see [ImageChangeTransferSyntax](#)) one operation is to make sure the [VR](#) of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the [VR](#) is not stored directly in the file.

Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the [VR](#) to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that [VR](#) is valid for the encoding
- One has to make sure that [VR](#) 16bits can store the original value length

Examples

[ExplicitLittleEndian.cs](#), [GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

12.130.2 Constructor & Destructor Documentation

12.130.2.1 FileExplicitFilter()

```
gdcm::FileExplicitFilter::FileExplicitFilter () [inline]
```

12.130.2.2 ~FileExplicitFilter()

```
gdcm::FileExplicitFilter::~~FileExplicitFilter () [default]
```

12.130.3 Member Function Documentation

12.130.3.1 Change()

```
bool gdcm::FileExplicitFilter::Change ()
```

Set FMI Transfer Syntax.

Change

Examples

[ExplicitLittleEndian.cs](#), [GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

12.130.3.2 ChangeFMI()

```
bool gdcmm::FileExplicitFilter::ChangeFMI () [protected]
```

12.130.3.3 GetFile()

```
File & gdcmm::FileExplicitFilter::GetFile () [inline]
```

Examples

[ExplicitLittleEndian.cs](#).

12.130.3.4 ProcessDataSet()

```
bool gdcmm::FileExplicitFilter::ProcessDataSet (  
    DataSet & ds,  
    Dicts const & dicts) [protected]
```

12.130.3.5 SetChangePrivateTags()

```
void gdcmm::FileExplicitFilter::SetChangePrivateTags (  
    bool b) [inline]
```

Decide whether or not to [VR](#)'ify private tags.

Examples

[ExplicitLittleEndian.cs](#).

12.130.3.6 SetFile()

```
void gdcmm::FileExplicitFilter::SetFile (  
    const File & f) [inline]
```

Set/Get [File](#).

Examples

[ExplicitLittleEndian.cs](#), [GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

12.130.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (  
    bool b)
```

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

12.130.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (  
    bool b)
```

12.130.3.9 SetUseVRUN()

```
void gdcm::FileExplicitFilter::SetUseVRUN (  
    bool b) [inline]
```

When [VR](#)=16bits in explicit but Implicit has a 32bits length, use [VR](#)=UN.

The documentation for this class was generated from the following file:

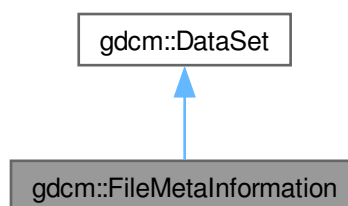
- [gdcmFileExplicitFilter.h](#)

12.131 gdcm::FileMetaInformation Class Reference

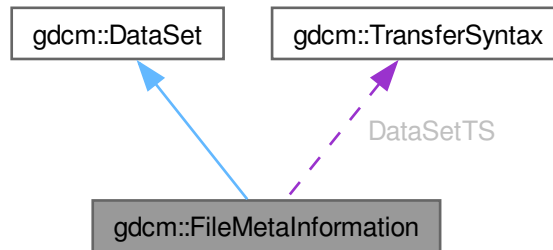
Class to represent a [File](#) Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for `gdcm::FileMetaInformation`:



Public Member Functions

- [FileMetaInformation](#) ()
- [FileMetaInformation](#) ([FileMetaInformation](#) const &[fmi](#))=default
- [~FileMetaInformation](#) ()
- void [FillFromDataSet](#) ([DataSet](#) const &[ds](#))
 - Construct a [FileMetaInformation](#) from an already existing [DataSet](#):
- const [TransferSyntax](#) & [GetDataSetTransferSyntax](#) () const
- [VL](#) [GetFullLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () const
- std::string [GetMediaStorageAsString](#) () const
- [TransferSyntax::NegociatedType](#) [GetMetaInformationTS](#) () const
- [Preamble](#) & [GetPreamble](#) ()
- const [Preamble](#) & [GetPreamble](#) () const
 - Get [Preamble](#).
- void [Insert](#) (const [DataElement](#) &[de](#))
- bool [IsValid](#) () const
- [FileMetaInformation](#) & [operator=](#) (const [FileMetaInformation](#) &[fmi](#))=default
- std::istream & [Read](#) (std::istream &[is](#))
 - Read.
- std::istream & [ReadCompat](#) (std::istream &[is](#))
- void [Replace](#) (const [DataElement](#) &[de](#))
- void [SetDataSetTransferSyntax](#) (const [TransferSyntax](#) &[ts](#))
- void [SetPreamble](#) (const [Preamble](#) &[p](#))
- std::ostream & [Write](#) (std::ostream &[os](#)) const
 - Write.

Public Member Functions inherited from [gdcm::DataSet](#)

- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()
- template<typename TDE>
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [PrivateTag](#) &t) const
Look up if private tag 't' is present in the dataset:
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [FindNextDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const
Return the dataelement.
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- [DataElementSet](#) & [GetDES](#) ()
- const [DataElementSet](#) & [GetDES](#) () const
- template<typename TDE>
[VL](#) [GetLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const
- [PrivateTag](#) [GetPrivateTag](#) (const [Tag](#) &t) const
Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsEmpty](#) () const
Returns if the dataset is empty.
- const [DataElement](#) & [operator\(\)](#) (uint16_t group, uint16_t element) const
- [DataSet](#) & [operator=](#) ([DataSet](#) const &)=default
- const [DataElement](#) & [operator\[\]](#) (const [Tag](#) &t) const
- void [Print](#) (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadNested](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTags](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedPrivateTagsWithLength](#) (std::istream &is, const std::set< [PrivateTag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTag](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadUpToTagWithLength](#) (std::istream &is, const [Tag](#) &t, std::set< [Tag](#) > const &skiptags, [VL](#) &length)

- `template<typename TDE, typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `SizeType Remove` (const `Tag` &tag)
 Completely remove a dataelement from the dataset.
- `void Replace` (const `DataElement` &de)
 Replace a dataelement with another one.
- `void ReplaceEmpty` (const `DataElement` &de)
 Only replace a DICOM attribute when it is missing or empty.
- `SizeType Size` () const
- `template<typename TDE, typename TSwap>`
`std::ostream const & Write` (std::ostream &os) const

Static Public Member Functions

- static void `AppendImplementationClassUID` (const char *imp)
- static const char * `GetImplementationClassUID` ()
- static const char * `GetImplementationVersionName` ()
- static const char * `GetSourceApplicationEntityTitle` ()
- static void `SetImplementationClassUID` (const char *imp)
 Override the GDCM default values:
- static void `SetImplementationVersionName` (const char *version)
- static void `SetSourceApplicationEntityTitle` (const char *title)

Protected Member Functions

- void `ComputeDataSetMediaStorageSOPClass` ()
- void `ComputeDataSetTransferSyntax` ()
- void `Default` ()
- `template<typename TSwap>`
`std::istream & ReadCompatInternal` (std::istream &is)

Protected Member Functions inherited from `gdcm::DataSet`

- `Tag ComputeDataElement` (const `PrivateTag` &t) const
- const `DataElement` & `GetDEEnd` () const
- void `InsertDataElement` (const `DataElement` &de)

Static Protected Member Functions

- static const char * `GetFileMetaInformationVersion` ()
- static const char * `GetGDCMImplementationClassUID` ()
- static const char * `GetGDCMImplementationVersionName` ()
- static const char * `GetGDCMSourceApplicationEntityTitle` ()

Protected Attributes

- [MediaStorage::MSType DataSetMS](#)
- [TransferSyntax DataSetTS](#)
- [TransferSyntax::NegociatedType MetaInformationTS](#)

Friends

- `std::ostream & operator<< (std::ostream &_os, const FileMetaInformation &_val)`

Additional Inherited Members

Public Types inherited from [gdcmm::DataSet](#)

- `typedef DataSet::const_iterator ConstIterator`
- `typedef std::set< DataElement > DataElementSet`
- `typedef DataSet::iterator Iterator`
- `typedef DataSet::size_type SizeType`

12.131.1 Detailed Description

Class to represent a [File](#) Meta Information.

[FileMetaInformation](#) is a Explicit Structured Set. Whenever the file contains an [ImplicitDataElement DataSet](#), a conversion will take place.

Definition: The [File](#) Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte [File Preamble](#), followed by a 4 byte DICOM prefix, followed by the [File](#) Meta Elements shown in [Table 7.1-1](#). This header shall be present in every DICOM file.

See also

[Writer Reader](#)

Examples

[DumpToshibaDTL.cxx](#), [DumpToshibaDTI2.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

12.131.2 Constructor & Destructor Documentation

12.131.2.1 [FileMetaInformation\(\)](#) [1/2]

`gdcmm::FileMetaInformation::FileMetaInformation ()`

Referenced by [FileMetaInformation\(\)](#), [~FileMetaInformation\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.131.2.2 ~FileMetaInformation()

gdcm::FileMetaInformation::~~FileMetaInformation ()

References [FileMetaInformation\(\)](#), and [operator<<](#).

12.131.2.3 FileMetaInformation() [2/2]

gdcm::FileMetaInformation::FileMetaInformation (
FileMetaInformation const & fmi) [default]

References [FileMetaInformation\(\)](#).

12.131.3 Member Function Documentation

12.131.3.1 AppendImplementationClassUID()

void gdcm::FileMetaInformation::AppendImplementationClassUID (
const char * imp) [static]

12.131.3.2 ComputeDataSetMediaStorageSOPClass()

void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass () [protected]

12.131.3.3 ComputeDataSetTransferSyntax()

void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax () [protected]

12.131.3.4 Default()

void gdcm::FileMetaInformation::Default () [protected]

12.131.3.5 FillFromDataSet()

void gdcm::FileMetaInformation::FillFromDataSet (
[DataSet](#) const & ds)

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

12.131.3.6 GetDataSetTransferSyntax()

const [TransferSyntax](#) & gdcM::FileMetaInformation::GetDataSetTransferSyntax () const [inline]

Examples

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

References [DataSetTS](#).

12.131.3.7 GetFileMetaInformationVersion()

const char * gdcM::FileMetaInformation::GetFileMetaInformationVersion () [static], [protected]

12.131.3.8 GetFullLength()

[VL](#) gdcM::FileMetaInformation::GetFullLength () const [inline]

References [gdcM::DataSet::GetLength\(\)](#), and [gdcM::VL::GetLength\(\)](#).

12.131.3.9 GetGDCMImplementationClassUID()

const char * gdcM::FileMetaInformation::GetGDCMImplementationClassUID () [static], [protected]

12.131.3.10 GetGDCMImplementationVersionName()

const char * gdcM::FileMetaInformation::GetGDCMImplementationVersionName () [static], [protected]

12.131.3.11 GetGDCMSourceApplicationEntityTitle()

const char * gdcM::FileMetaInformation::GetGDCMSourceApplicationEntityTitle () [static], [protected]

12.131.3.12 GetImplementationClassUID()

const char * gdcM::FileMetaInformation::GetImplementationClassUID () [static]

12.131.3.13 GetImplementationVersionName()

const char * gdcM::FileMetaInformation::GetImplementationVersionName () [static]

12.131.3.14 GetMediaStorage()

[MediaStorage](#) gdcm::FileMetaInformation::GetMediaStorage () const

12.131.3.15 GetMediaStorageAsString()

std::string gdcm::FileMetaInformation::GetMediaStorageAsString () const

12.131.3.16 GetMetaInformationTS()

[TransferSyntax::NegociatedType](#) gdcm::FileMetaInformation::GetMetaInformationTS () const [inline]

References [MetaInformationTS](#).

12.131.3.17 GetPreamble() [1/2]

[Preamble](#) & gdcm::FileMetaInformation::GetPreamble () [inline]

12.131.3.18 GetPreamble() [2/2]

const [Preamble](#) & gdcm::FileMetaInformation::GetPreamble () const [inline]

Get [Preamble](#).

Referenced by [operator<<](#).

12.131.3.19 GetSourceApplicationEntityTitle()

const char * gdcm::FileMetaInformation::GetSourceApplicationEntityTitle () [static]

12.131.3.20 Insert()

void gdcm::FileMetaInformation::Insert (
 const [DataElement](#) & de) [inline]

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DataElement::GetTag\(\)](#), and [gdcm::DataSet::InsertDataElement](#)

Referenced by [Replace\(\)](#).

12.131.3.21 IsValid()

bool gdcm::FileMetaInformation::IsValid () const [inline]

12.131.3.22 operator=()

[FileMetaInformation](#) & gdcmm::FileMetaInformation::operator= (
const [FileMetaInformation](#) & fmi) [default]

References [FileMetaInformation\(\)](#).

12.131.3.23 Read()

std::istream & gdcmm::FileMetaInformation::Read (
std::istream & is)

Read.

12.131.3.24 ReadCompat()

std::istream & gdcmm::FileMetaInformation::ReadCompat (
std::istream & is)

12.131.3.25 ReadCompatInternal()

template<typename TSwap>
std::istream & gdcmm::FileMetaInformation::ReadCompatInternal (
std::istream & is) [protected]

12.131.3.26 Replace()

void gdcmm::FileMetaInformation::Replace (
const [DataElement](#) & de) [inline]

Examples

[LargeVRDSExplicit.cxx](#).

References [gdcmm::DataElement::GetTag\(\)](#), [Insert\(\)](#), and [gdcmm::DataSet::Remove\(\)](#).

12.131.3.27 SetDataSetTransferSyntax()

void gdcmm::FileMetaInformation::SetDataSetTransferSyntax (
const [TransferSyntax](#) & ts)

Examples

[CreateJPIPDataSet.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [EncapsulateFileInRawData.cxx](#),
[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixJAIBugJPEGLS.cxx](#),
[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#),
[QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.131.3.28 SetImplementationClassUID()

```
void gdcmm::FileMetaInformation::SetImplementationClassUID (
    const char * imp) [static]
```

Override the GDCM default values:

12.131.3.29 SetImplementationVersionName()

```
void gdcmm::FileMetaInformation::SetImplementationVersionName (
    const char * version) [static]
```

12.131.3.30 SetPreamble()

```
void gdcmm::FileMetaInformation::SetPreamble (
    const Preamble & p) [inline]
```

12.131.3.31 SetSourceApplicationEntityTitle()

```
void gdcmm::FileMetaInformation::SetSourceApplicationEntityTitle (
    const char * title) [static]
```

Examples

[ExplicitLittleEndian.cs](#), [FixJAIBugJPEGLS.cxx](#), [GenerateDICOMDIR.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

12.131.3.32 Write()

```
std::ostream & gdcmm::FileMetaInformation::Write (
    std::ostream & os) const
```

Write.

12.131.4 Friends And Related Symbol Documentation

12.131.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const FileMetaInformation & __val) [friend]
```

References [FileMetaInformation\(\)](#), [GetPreamble\(\)](#), [operator<<](#), and [gdcmm::DataSet::Print\(\)](#).

Referenced by [~FileMetaInformation\(\)](#), and [operator<<](#).

12.131.5 Member Data Documentation

12.131.5.1 DataSetMS

[MediaStorage::MSType](#) gdcm::FileMetaInformation::DataSetMS [protected]

12.131.5.2 DataSetTS

[TransferSyntax](#) gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by [GetDataSetTransferSyntax\(\)](#).

12.131.5.3 MetaInformationTS

[TransferSyntax::NegociatedType](#) gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by [GetMetaInformationTS\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmFileMetaInformation.h](#)

12.132 gdcm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmFilename.h>
```

Public Member Functions

- [Filename](#) (const char *filename="")
- bool [EndWith](#) (const char ending[]) const
Does the filename ends with a particular string ?
- const char * [GetExtension](#) ()
return only the extension part of a filename
- const char * [GetFileName](#) () const
Return the full filename.
- const char * [GetName](#) ()
return only the name part of a filename
- const char * [GetPath](#) ()
Return only the path component of a filename.
- bool [IsEmpty](#) () const
return whether the filename is empty
- bool [IsIdentical](#) ([Filename](#) const &fn) const
- [operator const char *](#) () const
- const char * [ToUnixSlashes](#) ()
Convert backslash (windows style) to UNIX style slash.
- const char * [ToWindowsSlashes](#) ()
Convert forward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * [Join](#) (const char *path, const char *filename)

12.132.1 Detailed Description

Class to manipulate file name's.

Note

OS independent representation of a filename (to query path, name and extension from a filename)

12.132.2 Constructor & Destructor Documentation

12.132.2.1 Filename()

```
gdcmm::Filename::Filename (  
    const char * filename = "") [inline]
```

Referenced by [IsIdentical\(\)](#).

12.132.3 Member Function Documentation

12.132.3.1 EndWith()

```
bool gdcmm::Filename::EndWith (  
    const char ending[]) const
```

Does the filename ends with a particular string ?

12.132.3.2 GetExtension()

```
const char * gdcmm::Filename::GetExtension ()
```

return only the extension part of a filename

12.132.3.3 GetFileName()

```
const char * gdcmm::Filename::GetFileName () const [inline]
```

Return the full filename.

Referenced by [operator const char *\(\)](#).

12.132.3.4 GetName()

```
const char * gdcm::Filename::GetName ()
```

return only the name part of a filename

12.132.3.5 GetPath()

```
const char * gdcm::Filename::GetPath ()
```

Return only the path component of a filename.

Examples

```
ClinicalTrialIdentificationWorkflow.cs.
```

12.132.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty () const [inline]
```

return whether the filename is empty

12.132.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (  
    Filename const & fn) const
```

References [Filename\(\)](#).

12.132.3.8 Join()

```
const char * gdcm::Filename::Join (  
    const char * path,  
    const char * filename) [static]
```

Join two paths NOT THREAD SAFE

12.132.3.9 operator const char *()

```
gdcm::Filename::operator const char * () const [inline]
```

Simple operator to allow [Filename](#) myfilename("...") ; const char * s = myfilename;

References [GetFileName\(\)](#).

12.132.3.10 ToUnixSlashes()

```
const char * gdcm::Filename::ToUnixSlashes ()
```

Convert backslash (windows style) to UNIX style slash.

12.132.3.11 ToWindowsSlashes()

```
const char * gdcm::Filename::ToWindowsSlashes ()
```

Convert forward slash (UNIX style) to windows style slash.

The documentation for this class was generated from the following file:

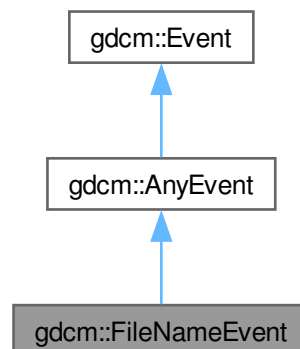
- [gdcmFilename.h](#)

12.133 gdcm::FileNameEvent Class Reference

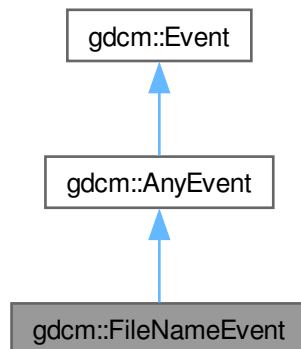
[FileNameEvent](#).

```
#include <gdcmFileNameEvent.h>
```

Inheritance diagram for gdcm::FileNameEvent:



Collaboration diagram for gdcm::FileNameEvent:



Public Types

- typedef [FileNameEvent](#) [Self](#)
- typedef [AnyEvent](#) [Superclass](#)

Public Member Functions

- [FileNameEvent](#) (const char *s="")
- [FileNameEvent](#) (const [Self](#) &s)
- [~FileNameEvent](#) () override=default
- bool [CheckEvent](#) (const [::gdcm::Event](#) *e) const override
- const char * [GetEventName](#) () const override
- const char * [GetFileName](#) () const
- [::gdcm::Event](#) * [MakeObject](#) () const override
- void [operator=](#) (const [Self](#) &)=delete
- void [SetFileName](#) (const char *f)

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const [Event](#) &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

12.133.1 Detailed Description

[FileNameEvent](#).

Special type of event triggered during processing of [FileSet](#)

See also

[AnyEvent](#)

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.133.2 Member Typedef Documentation

12.133.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

12.133.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

12.133.3 Constructor & Destructor Documentation

12.133.3.1 [FileNameEvent\(\)](#) [1/2]

```
gdcm::FileNameEvent::FileNameEvent (  
    const char * s = "") [inline]
```

12.133.3.2 [~FileNameEvent\(\)](#)

```
gdcm::FileNameEvent::~~FileNameEvent () [override], [default]
```

12.133.3.3 [FileNameEvent\(\)](#) [2/2]

```
gdcm::FileNameEvent::FileNameEvent (  
    const Self & s) [inline]
```


12.133.4 Member Function Documentation

12.133.4.1 CheckEvent()

```
bool gdcmm::FileNameEvent::CheckEvent (
    const gdcmm::Event * e) const    [inline], [override]
```

12.133.4.2 GetEventName()

```
const char * gdcmm::FileNameEvent::GetEventName () const    [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

12.133.4.3 GetFileName()

```
const char * gdcmm::FileNameEvent::GetFileName () const    [inline]
```

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.133.4.4 MakeObject()

```
gdcmm::Event * gdcmm::FileNameEvent::MakeObject () const    [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcmm::Event](#).

12.133.4.5 operator=()

```
void gdcmm::FileNameEvent::operator= (
    const Self & )    [delete]
```

12.133.4.6 SetFileName()

```
void gdcmm::FileNameEvent::SetFileName (
    const char * f)    [inline]
```

The documentation for this class was generated from the following file:

- [gdcmmFileNameEvent.h](#)

12.134 gdcm::FilenameGenerator Class Reference

[FilenameGenerator](#).

```
#include <gdcmFilenameGenerator.h>
```

Public Types

- typedef std::vector< [FilenameType](#) > [FileNamesType](#)
- typedef std::string [FilenameType](#)
- typedef [FileNamesType](#)::size_type [SizeType](#)

Public Member Functions

- [FilenameGenerator](#) ()
- [~FilenameGenerator](#) ()=default
- bool [Generate](#) ()
Generate (return success).
- const char * [GetFilename](#) ([SizeType](#) n) const
Get a particular filename (call after Generate).
- [FileNamesType](#) const & [GetFileNames](#) () const
Return all filenames.
- [SizeType](#) [GetNumberOfFileNames](#) () const
- const char * [GetPattern](#) () const
- const char * [GetPrefix](#) () const
- void [SetNumberOfFileNames](#) ([SizeType](#) nfiles)
Set/Get the number of filenames to generate.
- void [SetPattern](#) (const char *pattern)
Set/Get pattern.
- void [SetPrefix](#) (const char *prefix)
Set/Get prefix.

12.134.1 Detailed Description

[FilenameGenerator](#).

class to generate filenames based on a pattern (C-style)

Output will be:

for i = 0, number of filenames: outfilename[i] = prefix + (pattern % i)

where pattern % i means C-style sprintf of Pattern using value 'i'

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.2 Member Typedef Documentation

12.134.2.1 FilenamesType

typedef std::vector<[FilenameType](#)> [gdcm::FilenameGenerator::FilenamesType](#)

12.134.2.2 FilenameType

typedef std::string [gdcm::FilenameGenerator::FilenameType](#)

12.134.2.3 SizeType

typedef FilenamesType::size_type [gdcm::FilenameGenerator::SizeType](#)

12.134.3 Constructor & Destructor Documentation

12.134.3.1 FilenameGenerator()

[gdcm::FilenameGenerator::FilenameGenerator \(\)](#) [inline]

12.134.3.2 ~FilenameGenerator()

[gdcm::FilenameGenerator::~~FilenameGenerator \(\)](#) [default]

12.134.4 Member Function Documentation

12.134.4.1 Generate()

bool [gdcm::FilenameGenerator::Generate \(\)](#)

Generate (return success).

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.4.2 GetFilename()

```
const char * gdcm::FilenameGenerator::GetFilename (  
    SizeType n) const
```

Get a particular filename (call after Generate).

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.4.3 GetFileNames()

```
FileNamesType const & gdcm::FilenameGenerator::GetFileNames () const [inline]
```

Return all filenames.

References [gdcm_assert](#).

12.134.4.4 GetNumberOfFileNames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFileNames () const
```

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.4.5 GetPattern()

```
const char * gdcm::FilenameGenerator::GetPattern () const [inline]
```

12.134.4.6 GetPrefix()

```
const char * gdcm::FilenameGenerator::GetPrefix () const [inline]
```

12.134.4.7 SetNumberOfFileNames()

```
void gdcm::FilenameGenerator::SetNumberOfFileNames (  
    SizeType nfiles)
```

Set/Get the number of filenames to generate.

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (  
    const char * pattern) [inline]
```

Set/Get pattern.

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.134.4.9 SetPrefix()

```
void gdcm::FilenameGenerator::SetPrefix (  
    const char * prefix) [inline]
```

Set/Get prefix.

The documentation for this class was generated from the following file:

- [gdcmFilenameGenerator.h](#)

12.135 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< [FileType](#) > [FilesType](#)
- typedef std::string [FileType](#)

Public Member Functions

- [FileSet](#) ()
- bool [AddFile](#) (const char *filename)
- void [AddFile](#) ([File](#) const &)
- [FilesType](#) const & [GetFiles](#) () const
- void [SetFiles](#) ([FileType](#) const &files)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [FileSet](#) &d)

12.135.1 Detailed Description

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which [File](#) IDs are unique.

12.135.2 Member Typedef Documentation

12.135.2.1 FileType

```
typedef std::vector<FileType> gdcm::FileSet::FileType
```

12.135.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

12.135.3 Constructor & Destructor Documentation

12.135.3.1 FileSet()

```
gdcm::FileSet::FileSet () [inline]
```

Referenced by [operator<<](#).

12.135.4 Member Function Documentation

12.135.4.1 AddFile() [1/2]

```
bool gdcm::FileSet::AddFile (  
    const char * filename)
```

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

12.135.4.2 AddFile() [2/2]

```
void gdcm::FileSet::AddFile (  
    File const & ) [inline]
```

[Deprecated](#) . Does nothing

12.135.4.3 GetFiles()

[FileType](#) const & gdcm::FileSet::GetFiles () const [inline]

12.135.4.4 SetFiles()

void gdcm::FileSet::SetFiles (
 [FileType](#) const & files)

12.135.5 Friends And Related Symbol Documentation

12.135.5.1 operator<<

std::ostream & operator<< (
 std::ostream & _os,
 const [FileSet](#) & d) [friend]

References [FileSet\(\)](#).

The documentation for this class was generated from the following file:

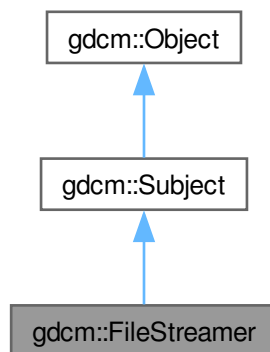
- [gdcmFileSet.h](#)

12.136 gdcm::FileStreamer Class Reference

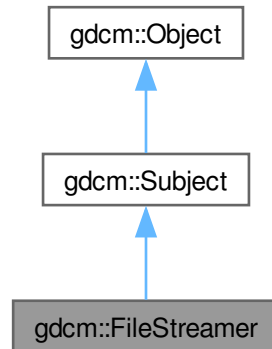
[FileStreamer](#).

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for `gdcm::FileStreamer`:



Public Member Functions

- [FileStreamer](#) ()
- [~FileStreamer](#) () override
- bool [AppendToDataElement](#) (const [Tag](#) &t, const char *array, size_t len)
Append to previously started [Tag](#) t.
- bool [AppendToGroupDataElement](#) (const [PrivateTag](#) &pt, const char *array, size_t len)
Append to previously started private creator.
- bool [CheckDataElement](#) (const [Tag](#) &t)
- void [CheckTemplateFileName](#) (bool check)
- bool [ReserveDataElement](#) (size_t len)
- bool [ReserveGroupDataElement](#) (unsigned short ndataelement)
- void [SetOutputFileName](#) (const char *filename__native)
Set output filename (target file).
- void [SetTemplateFileName](#) (const char *filename__native)
Set input DICOM template filename.
- bool [StartDataElement](#) (const [Tag](#) &t)
- bool [StartGroupDataElement](#) (const [PrivateTag](#) &pt, size_t maxsize=0, uint8_t startoffset=0)
- bool [StopDataElement](#) (const [Tag](#) &t)
Stop appending to tag t. This will compute the proper attribute length.
- bool [StopGroupDataElement](#) (const [PrivateTag](#) &pt)
Stop appending to private creator.

Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)

- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [FileStreamer](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.136.1 Detailed Description

[FileStreamer](#).

This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

This class support two mode of operation:

1. Creating a single [DataElement](#) by appending chunk after chunk of data.
2. Creating a set of [DataElement](#) within the same group, using a private creator for start. New [DataElement](#) are added any time the user defined maximum size for data element is reached.

Warning

any existing [DataElement](#) is removed, pick carefully which [DataElement](#) to add.

Examples

[FileStreaming.cs](#).

12.136.2 Constructor & Destructor Documentation

12.136.2.1 FileStreamer()

gdcm::FileStreamer::FileStreamer ()

Referenced by [New\(\)](#).

12.136.2.2 ~FileStreamer()

gdcm::FileStreamer::~~FileStreamer () [override]

12.136.3 Member Function Documentation

12.136.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
    const Tag & t,
    const char * array,
    size_t len)
```

Append to previously started [Tag](#) t.

12.136.3.2 AppendToGroupDataElement()

```
bool gdcm::FileStreamer::AppendToGroupDataElement (
    const PrivateTag & pt,
    const char * array,
    size_t len)
```

Append to previously started private creator.

Examples

[FileStreaming.cs](#).

12.136.3.3 CheckDataElement()

```
bool gdcm::FileStreamer::CheckDataElement (
    const Tag & t)
```

Decide to check the Data [Element](#) to be written (default: off) The implementation has default strategy for checking validity of [DataElement](#). Currently it only support checking for the following tags:

- (7fe0,0010) Pixel Data

12.136.3.4 CheckTemplateFileName()

```
void gdcm::FileStreamer::CheckTemplateFileName (  
    bool check)
```

Instead of simply blindly copying the input DICOM Template file, GDCM will be used to check the input file, and correct any issues recognized within the file. Only use if you do not have control over the input template file.

12.136.3.5 New()

```
SmartPointer< FileStreamer > gdcm::FileStreamer::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References [FileStreamer\(\)](#).

12.136.3.6 ReserveDataElement()

```
bool gdcm::FileStreamer::ReserveDataElement (  
    size_t len)
```

Add a hint on the final size of the dataelement. When optimally chosen, this reduce the number of file in-place copying. Should be called before `StartDataElement`

12.136.3.7 ReserveGroupDataElement()

```
bool gdcm::FileStreamer::ReserveGroupDataElement (  
    unsigned short ndataelement)
```

Optimisation: pre-allocate the number of dataelement within the private group (ndataelement <= 256). Should be called before `StartGroupDataElement`

12.136.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (  
    const char * filename__native)
```

Set output filename (target file).

Examples

[FileStreaming.cs](#).

12.136.3.9 SetTemplateFileName()

```
void gdcmm::FileStreamer::SetTemplateFileName (
    const char * filename_native)
```

Set input DICOM template filename.

Examples

[FileStreaming.cs](#).

12.136.3.10 StartDataElement()

```
bool gdcmm::FileStreamer::StartDataElement (
    const Tag & t)
```

Start Single Data [Element](#) Operation This will delete any existing [Tag](#) t. Need to call it only once.

12.136.3.11 StartGroupDataElement()

```
bool gdcmm::FileStreamer::StartGroupDataElement (
    const PrivateTag & pt,
    size_t maxsize = 0,
    uint8_t startoffset = 0)
```

Start Private Group (multiple [DataElement](#)) Operation. Each newly added [DataElement](#) will have a length lower than

Parameters

maxsize	. When not specified, maxsize is set to maximum size allowed by DICOM (= 2 ³²). startoffset can be used to specify the very first element you want to start with (instead of the first possible). Value should be in [0x0, 0xff] This will find the first available private creator.
---------	--

[Bug](#) maxsize should be a value lower than the actual total size of the buffer to be copied

Examples

[FileStreaming.cs](#).

12.136.3.12 StopDataElement()

```
bool gdcmm::FileStreamer::StopDataElement (
    const Tag & t)
```

Stop appending to tag t. This will compute the proper attribute length.

12.136.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (  
    const PrivateTag & pt)
```

Stop appending to private creator.

Examples

[FileStreaming.cs](#).

The documentation for this class was generated from the following file:

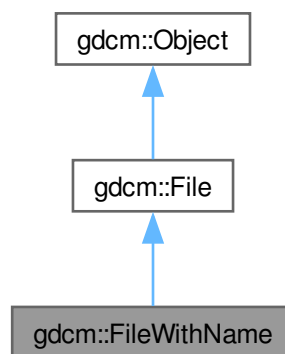
- [gdcmFileStreamer.h](#)

12.137 gdcm::FileWithName Class Reference

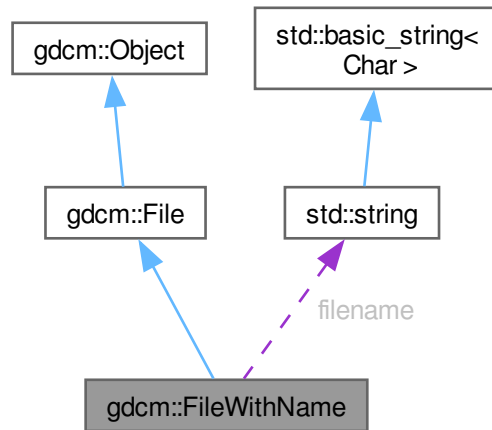
[FileWithName](#).

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for `gdcm::FileWithName`:



Public Member Functions

- [FileWithName](#) ([File](#) &[f](#))

Public Member Functions inherited from [gdcm::File](#)

- [File](#) ()
- [~File](#) () override
- [DataSet](#) & [GetDataSet](#) ()
Get Data Set.
- const [DataSet](#) & [GetDataSet](#) () const
Get Data Set.
- [FileMetaInformation](#) & [GetHeader](#) ()
Get [File](#) Meta Information.
- const [FileMetaInformation](#) & [GetHeader](#) () const
Get [File](#) Meta Information.
- `std::istream` & [Read](#) (`std::istream` &[is](#))
Read.
- void [SetDataSet](#) (const [DataSet](#) &[ds](#))
Set Data Set.
- void [SetHeader](#) (const [FileMetaInformation](#) &[fmi](#))
Set [File](#) Meta Information.
- `std::ostream` const & [Write](#) (`std::ostream` &[os](#)) const
Write.

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Public Attributes

- std::string [filename](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.137.1 Detailed Description

[FileWithName](#).

Backward only class do not use in newer code

12.137.2 Constructor & Destructor Documentation

12.137.2.1 FileWithName()

```
gdcm::FileWithName::FileWithName (  
    File & f) [inline]
```

References [gdcm::File::File\(\)](#), and [filename](#).

12.137.3 Member Data Documentation

12.137.3.1 filename

std::string gdcm::FileWithName::filename

Referenced by [FileWithName\(\)](#).

The documentation for this class was generated from the following file:

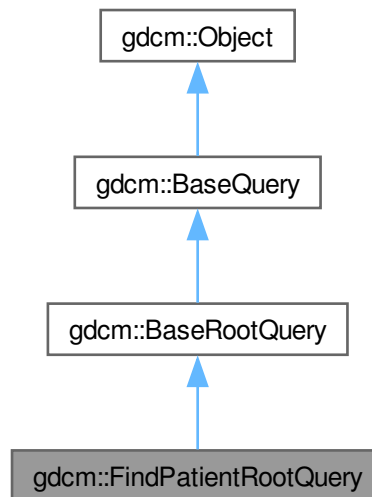
- [gdcmSerieHelper.h](#)

12.138 gdc::FindPatientRootQuery Class Reference

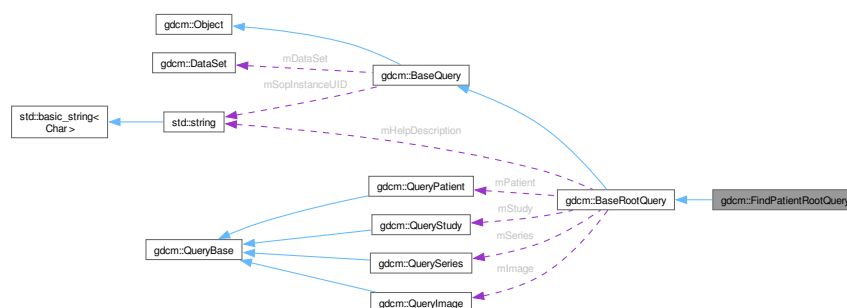
PatientRootQuery.

```
#include <gdcFindPatientRootQuery.h>
```

Inheritance diagram for gdc::FindPatientRootQuery:



Collaboration diagram for gdc::FindPatientRootQuery:



Public Member Functions

- [FindPatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- `std::vector< Tag >` [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) * [Construct](#) (ERootType inRootType, EQueryLevel qlvel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) (EQueryLevel ql)

Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.138.1 Detailed Description

[PatientRootQuery](#).

contains: the class which will produce a dataset for c-find with patient root

12.138.2 Constructor & Destructor Documentation

12.138.2.1 [FindPatientRootQuery](#)()

[gdcm::FindPatientRootQuery::FindPatientRootQuery](#) ()

12.138.3 Member Function Documentation

12.138.3.1 [GetAbstractSyntaxUID](#)()

[UIDs::TSName](#) [gdcm::FindPatientRootQuery::GetAbstractSyntaxUID](#) () const [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.138.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindPatientRootQuery::GetTagListByLevel (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

12.138.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements [gdcm::BaseRootQuery](#).

12.138.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (  
    bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

12.138.4 Friends And Related Symbol Documentation

12.138.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

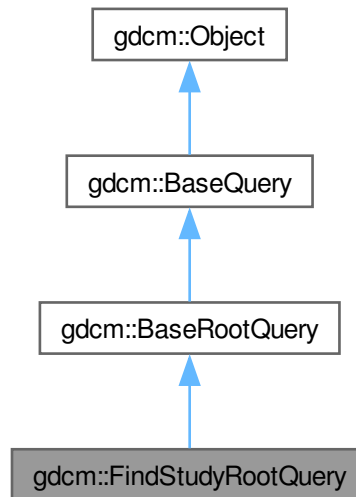
- [gdcmFindPatientRootQuery.h](#)

12.139 gdcM::FindStudyRootQuery Class Reference

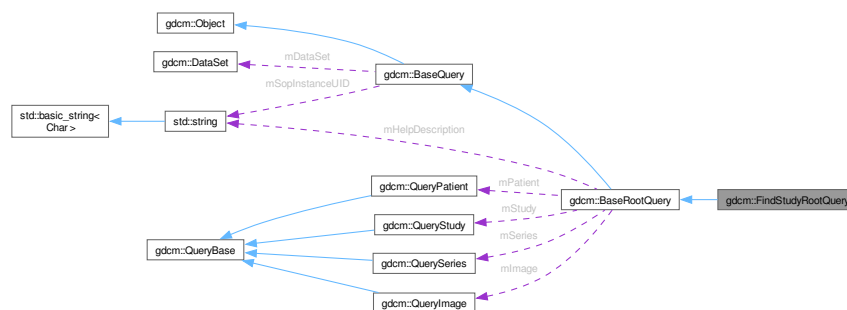
[FindStudyRootQuery](#).

```
#include <gdcMFindStudyRootQuery.h>
```

Inheritance diagram for gdcM::FindStudyRootQuery:



Collaboration diagram for gdcM::FindStudyRootQuery:



Public Member Functions

- [FindStudyRootQuery](#) ()
- `UIDs::TSName GetAbstractSyntaxUID` () const override
- `std::vector< Tag > GetTagListByLevel` (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) * [Construct](#) (ERootType inRootType, EQueryLevel qlvel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) (EQueryLevel ql)

Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &in↵ Value)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.139.1 Detailed Description

[FindStudyRootQuery](#).

contains: the class which will produce a dataset for C-FIND with study root

12.139.2 Constructor & Destructor Documentation

12.139.2.1 [FindStudyRootQuery](#)()

[gdcm::FindStudyRootQuery::FindStudyRootQuery](#) ()

12.139.3 Member Function Documentation

12.139.3.1 [GetAbstractSyntaxUID](#)()

[UIDs::TSName](#) [gdcm::FindStudyRootQuery::GetAbstractSyntaxUID](#) () const [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.139.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindStudyRootQuery::GetTagListByLevel (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

12.139.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements [gdcm::BaseRootQuery](#).

12.139.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (  
    bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that (0008,0052) is set that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional)

Implements [gdcm::BaseRootQuery](#).

12.139.4 Friends And Related Symbol Documentation

12.139.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

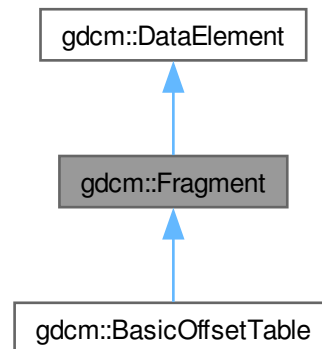
- [gdcmFindStudyRootQuery.h](#)

12.140 gdcmm::Fragment Class Reference

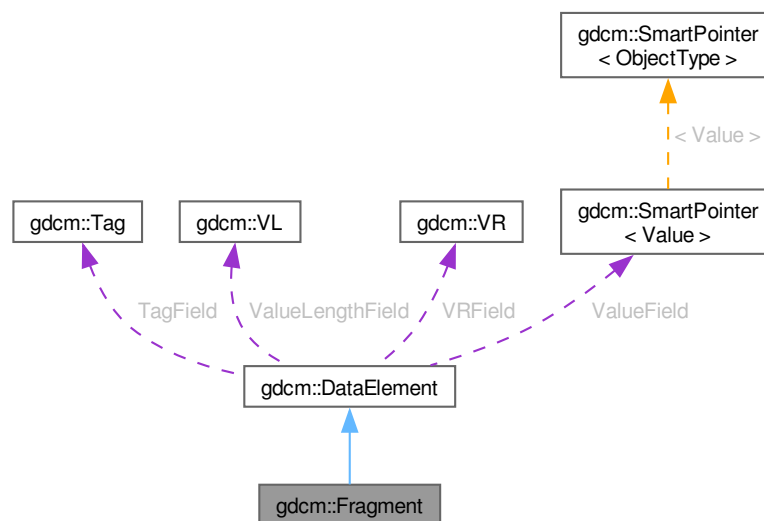
Class to represent a [Fragment](#).

```
#include <gdcmmFragment.h>
```

Inheritance diagram for gdcmm::Fragment:



Collaboration diagram for gdcmm::Fragment:



Public Member Functions

- [Fragment](#) ()
- [VL ComputeLength](#) () const
- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadBacktrack](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is)
- template<typename TSwap>
std::ostream & [Write](#) (std::ostream &os) const

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)

- `template<typename TDE, typename TSwap>`
`std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE, typename TSwap>`
`const std::ostream & Write (std::ostream &os) const`

Friends

- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.140.1 Detailed Description

Class to represent a [Fragment](#).

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#),
and [MpegVideoInfo.cs](#).

12.140.2 Constructor & Destructor Documentation

12.140.2.1 Fragment()

gdcm::Fragment::Fragment () [inline]

References [gdcm::DataElement::DataElement\(\)](#).

Referenced by [gdcm::BasicOffsetTable::BasicOffsetTable\(\)](#), and [operator<<](#).

12.140.3 Member Function Documentation

12.140.3.1 ComputeLength()

[VL](#) gdcm::Fragment::ComputeLength () const

12.140.3.2 GetLength()

[VL](#) gdcm::Fragment::GetLength () const

12.140.3.3 Read()

```
template<typename TSwap>
std::istream & gdcm::Fragment::Read (
    std::istream & is) [inline]
```

References [ReadPreValue\(\)](#), and [ReadValue\(\)](#).

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#).

12.140.3.4 ReadBacktrack()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadBacktrack (
    std::istream & is) [inline]
```

References [gdcm_assert](#), [gdcmErrorMacro](#), [gdcmWarningMacro](#), [gdcm::ParseException::SetLastElement\(\)](#), [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#).

12.140.3.5 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadPreValue (
    std::istream & is) [inline]
```

References [gdcm::DataElement::TagField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [Read\(\)](#).

12.140.3.6 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadValue (
    std::istream & is) [inline]
```

References [gdcmWarningMacro](#), [gdcm::ParseException::SetLastElement\(\)](#), [gdcm::DataElement::ValueField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [Read\(\)](#).

12.140.3.7 Write()

```
template<typename TSwap>
std::ostream & gdcm::Fragment::Write (
    std::ostream & os) const [inline]
```

References [gdcm::ByteValue::ComputeLength\(\)](#), [gdcm_assert](#), [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::ByteValue::GetLength\(\)](#), [gdcm::DataElement::IsEmpty\(\)](#), [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueLengthField](#), [gdcm::ByteValue::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

12.140.4 Friends And Related Symbol Documentation

12.140.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Fragment & val) [friend]
```

References [Fragment\(\)](#), [operator<<](#), [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [operator<<](#).

The documentation for this class was generated from the following file:

- [gdcmFragment.h](#)

12.141 gdcm::Global Class Reference

[Global](#).

```
#include <gdcmGlobal.h>
```

Public Member Functions

- [Global](#) ()
- [Global](#) (const [Global](#) &_val)=delete
- [~Global](#) ()
- bool [Append](#) (const char *path)
- [Defs](#) const & [GetDefs](#) () const
- [Dicts](#) & [GetDicts](#) ()
- [Dicts](#) const & [GetDicts](#) () const
- bool [LoadResourcesFiles](#) ()
- [Global](#) & [operator=](#) (const [Global](#) &_val)=delete
- bool [Prepend](#) (const char *path)

Static Public Member Functions

- static [Global](#) & [GetInstance](#) ()
return the singleton instance

Protected Member Functions

- const char * [Locate](#) (const char *resfile) const
Locate a resource file.

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Global](#) &g)

12.141.1 Detailed Description

[Global](#).

Note

[Global](#) should be included in any translation unit that will use [Dict](#) or that implements the singleton pattern. It makes sure that the [Dict](#) singleton is created before and destroyed after all other singletons in GDCM.

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

12.141.2 Constructor & Destructor Documentation

12.141.2.1 Global() [1/2]

gdcmm::Global::Global ()

Referenced by [Global\(\)](#), [GetInstance\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.141.2.2 ~Global()

gdcmm::Global::~~Global ()

12.141.2.3 Global() [2/2]

gdcmm::Global::Global (
 const Global & __val) [delete]

References [Global\(\)](#).

12.141.3 Member Function Documentation

12.141.3.1 Append()

bool gdcmm::Global::Append (
 const char * path)

Append path at the end of the path list

Warning

not thread safe !

12.141.3.2 GetDefs()

[Defs](#) const & gdcmm::Global::GetDefs () const

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples

[GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

12.141.3.3 GetDicts() [1/2]

[Dicts](#) & gdcmm::Global::GetDicts ()

12.141.3.4 GetDicts() [2/2]

[Dicts](#) const & gdcmm::Global::GetDicts () const

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

12.141.3.5 GetInstance()

[Global](#) & gdcmm::Global::GetInstance () [static]

return the singleton instance

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

References [Global\(\)](#).

12.141.3.6 LoadResourcesFiles()

bool gdcmm::Global::LoadResourcesFiles ()

Load all internal XML files, resource path need to have been set before calling this member function (see Append/Prepend members func)

Warning

not thread safe !

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

12.141.3.7 Locate()

```
const char * gdcmm::Global::Locate (
    const char * resfile) const    [protected]
```

Locate a resource file.

12.141.3.8 operator=()

```
Global & gdcmm::Global::operator= (
    const Global & _val)    [delete]
```

References [Global\(\)](#).

12.141.3.9 Prepend()

```
bool gdcmm::Global::Prepend (
    const char * path)
```

Prepend path at the beginning of the path list

Warning

not thread safe !

12.141.4 Friends And Related Symbol Documentation

12.141.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Global & g)    [friend]
```

References [Global\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmGlobal.h](#)

12.142 gdcmm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmmGroupDict.h>
```


Public Types

- typedef std::vector< std::string > [GroupStringVector](#)

Public Member Functions

- [GroupDict](#) ()
- [~GroupDict](#) ()=default
- std::string const & [GetAbbreviation](#) (uint16_t num) const
- std::string const & [GetName](#) (uint16_t num) const
- size_t [Size](#) () const

Protected Member Functions

- void [Add](#) (std::string const &abbreviation, std::string const &name)
- void [Insert](#) (uint16_t num, std::string const &abbreviation, std::string const &name)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [GroupDict](#) &_val)

12.142.1 Detailed Description

Class to represent the mapping from group number to its abbreviation and name.

Note

Should I rewrite this class to use a std::map instead of std::vector for problem of memory consumption ?

12.142.2 Member Typedef Documentation

12.142.2.1 GroupStringVector

typedef std::vector<std::string> [gdcn::GroupDict::GroupStringVector](#)

12.142.3 Constructor & Destructor Documentation

12.142.3.1 GroupDict()

[gdcn::GroupDict::GroupDict](#) () [inline]

Referenced by [~GroupDict\(\)](#), [Insert\(\)](#), and [operator<<](#).

12.142.3.2 ~GroupDict()

gdcmm::GroupDict::~~GroupDict () [default]

References [GroupDict\(\)](#), and [operator<<](#).

12.142.4 Member Function Documentation

12.142.4.1 Add()

```
void gdcmm::GroupDict::Add (  
    std::string const & abbreviation,  
    std::string const & name) [protected]
```

12.142.4.2 GetAbbreviation()

```
std::string const & gdcmm::GroupDict::GetAbbreviation (  
    uint16_t num) const
```

Referenced by [operator<<](#).

12.142.4.3 GetName()

```
std::string const & gdcmm::GroupDict::GetName (  
    uint16_t num) const
```

Referenced by [operator<<](#).

12.142.4.4 Insert()

```
void gdcmm::GroupDict::Insert (  
    uint16_t num,  
    std::string const & abbreviation,  
    std::string const & name) [protected]
```

References [GroupDict\(\)](#).

12.142.4.5 Size()

```
size_t gdcmm::GroupDict::Size () const [inline]
```

References [gdcmm_assert](#).

Referenced by [operator<<](#).

12.142.5 Friends And Related Symbol Documentation

12.142.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const GroupDict & _val) [friend]
```

References [GroupDict\(\)](#), [GetAbbreviation\(\)](#), [GetName\(\)](#), and [Size\(\)](#).

Referenced by [~GroupDict\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmGroupDict.h](#)

12.143 gdcm::IconImageFilter Class Reference

[IconImageFilter](#).

```
#include <gdcmIconImageFilter.h>
```

Public Member Functions

- [IconImageFilter](#) ()
- [~IconImageFilter](#) ()
- bool [Extract](#) ()
 - Extract all Icon found in [File](#).
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [IconImage](#) & [GetIconImage](#) (unsigned int i) const
- unsigned int [GetNumberOfIconImages](#) () const
 - Retrieve extract [IconImage](#) (need to call Extract first).
- void [SetFile](#) (const [File](#) &f)
 - Set/Get [File](#).

Protected Member Functions

- void [ExtractIconImages](#) ()
- void [ExtractVeprolIconImages](#) ()

12.143.1 Detailed Description

[IconImageFilter](#).

This filter will extract icons from a [File](#) This filter will loop over all known sequence (public and private) that may contains an [IconImage](#) and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12

Implementation details: This filter supports the following Icons:

- (0088,0200) Icon [Image](#) Sequence
- (0009,10,GEIIS) GE IIS Thumbnail Sequence
- (6003,10,GEMS_Ultrasound_ImageGroup_001) GEMS [Image](#) Thumbnail Sequence
- (0055,30,VEPRO VIF 3.0 DATA) Icon Data
- (0055,30,VEPRO VIM 5.0 DATA) ICONDATA2

Warning

the icon stored in those private attribute do not conform to definition of Icon [Image](#) Sequence (do not simply copy/paste). For example some private icon can be expressed as 12bits pixel, while the DICOM standard only allow 8bits icons.

See also

[ImageReader](#)

Examples

[ExtractIconFromFile.cxx](#).

12.143.2 Constructor & Destructor Documentation

12.143.2.1 [IconImageFilter](#)()

gdcm::IconImageFilter::IconImageFilter ()

12.143.2.2 [~IconImageFilter](#)()

gdcm::IconImageFilter::~~IconImageFilter ()

12.143.3 Member Function Documentation

12.143.3.1 Extract()

bool gdcm::IconImageFilter::Extract ()

Extract all Icon found in [File](#).

Examples

[ExtractIconFromFile.cxx](#).

12.143.3.2 ExtractIconImages()

void gdcm::IconImageFilter::ExtractIconImages () [protected]

12.143.3.3 ExtractVeproIconImages()

void gdcm::IconImageFilter::ExtractVeproIconImages () [protected]

12.143.3.4 GetFile() [1/2]

[File](#) & gdcm::IconImageFilter::GetFile () [inline]

12.143.3.5 GetFile() [2/2]

const [File](#) & gdcm::IconImageFilter::GetFile () const [inline]

12.143.3.6 GetIconImage()

[IconImage](#) & gdcm::IconImageFilter::GetIconImage (
unsigned int i) const

Examples

[ExtractIconFromFile.cxx](#).

12.143.3.7 GetNumberOfIconImages()

unsigned int gdcmm::IconImageFilter::GetNumberOfIconImages () const

Retrieve extract [IconImage](#) (need to call Extract first).

Examples

[ExtractIconFromFile.cxx](#).

12.143.3.8 SetFile()

void gdcmm::IconImageFilter::SetFile (
const [File](#) & f) [inline]

Set/Get [File](#).

Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmmIconImageFilter.h](#)

12.144 gdcmm::IconImageGenerator Class Reference

[IconImageGenerator](#).

```
#include <gdcmmIconImageGenerator.h>
```

Public Member Functions

- [IconImageGenerator](#) ()
- [~IconImageGenerator](#) ()
- void [AutoPixelMinMax](#) (bool b)
- void [ConvertRGBToPaletteColor](#) (bool b)
- bool [Generate](#) ()
Generate Icon.
- const [IconImage](#) & [GetIconImage](#) () const
Retrieve generated Icon.
- [Pixmap](#) & [GetPixmap](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
- void [SetOutputDimensions](#) (const unsigned int dims[2])
Set Target dimension of output Icon.
- void [SetOutsideValuePixel](#) (double v)
- void [SetPixelMinMax](#) (double min, double max)
- void [SetPixmap](#) (const [Pixmap](#) &p)
Set/Get [File](#).

12.144.1 Detailed Description

[IconImageGenerator](#).

This filter will generate a valid Icon from the Pixel Data element (an instance of [Pixmap](#)). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- MONOCHROME1
- MONOCHROME2
- PALETTE_COLOR

The Pixel Bits Allocated is restricted to 8bits, therefore 16 bits image needs to be rescaled. By default the filter will use the full scalar range of 16bits image to rescale to unsigned 8bits. This may not be ideal for some situation, in which case the API SetPixelMinMax can be used to overwrite the default min,max interval used.

See also

[ImageReader](#)

Examples

[ExtractIconFromFile.cxx](#).

12.144.2 Constructor & Destructor Documentation

12.144.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ()
```

12.144.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~~IconImageGenerator ()
```

12.144.3 Member Function Documentation

12.144.3.1 AutoPixelMinMax()

```
void gdcm::IconImageGenerator::AutoPixelMinMax (  
    bool b)
```

Instead of explicitly specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples

[ExtractIconFromFile.cxx](#).

12.144.3.2 ConvertRGBToPaletteColor()

```
void gdcm::IconImageGenerator::ConvertRGBToPaletteColor (
    bool b)
```

Converting from RGB to PALETTE_COLOR can be a slow operation. However DICOM standard requires that color icon be described as palette. Set this boolean to false only if you understand the consequences. default value is true, false generates invalid Icon [Image](#) Sequence

12.144.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ()
```

Generate Icon.

Examples

[ExtractIconFromFile.cxx](#).

12.144.3.4 GetIconImage()

```
const IconImage & gdcm::IconImageGenerator::GetIconImage () const [inline]
```

Retrieve generated Icon.

Examples

[ExtractIconFromFile.cxx](#).

12.144.3.5 GetPixmap() [1/2]

```
Pixmap & gdcm::IconImageGenerator::GetPixmap () [inline]
```

12.144.3.6 GetPixmap() [2/2]

```
const Pixmap & gdcm::IconImageGenerator::GetPixmap () const [inline]
```

12.144.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
    const unsigned int dims[2])
```

Set Target dimension of output Icon.

Examples

[ExtractIconFromFile.cxx](#).

12.144.3.8 SetOutsideValuePixel()

```
void gdcm::IconImageGenerator::SetOutsideValuePixel (  
    double v)
```

Set a pixel value that should be discarded. This happen typically for CT image, where a pixel has been used to pad outside the image (see Pixel Padding [Value](#)). Requires `AutoPixelMinMax(true)`

12.144.3.9 SetPixelMinMax()

```
void gdcm::IconImageGenerator::SetPixelMinMax (  
    double min,  
    double max)
```

Override default min/max to compute best rescale for 16bits -> 8bits downscale. Typically those value can be read from the `SmallestImagePixelValue` `LargestImagePixelValue` DICOM attribute.

12.144.3.10 SetPixmap()

```
void gdcm::IconImageGenerator::SetPixmap (  
    const Pixmap & p) [inline]
```

Set/Get [File](#).

Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmIconImageGenerator.h](#)

12.145 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- [ignore_char](#) (char c)

Public Attributes

- char [m_char](#)

12.145.1 Constructor & Destructor Documentation

12.145.1.1 ignore_char()

```
gdcm::ignore_char::ignore_char (  
    char c)    [inline]
```

References [m_char](#).

12.145.2 Member Data Documentation

12.145.2.1 m_char

```
char gdcm::ignore_char::m_char
```

Referenced by [ignore_char\(\)](#), and [gdcm::operator>>\(\)](#).

The documentation for this struct was generated from the following file:

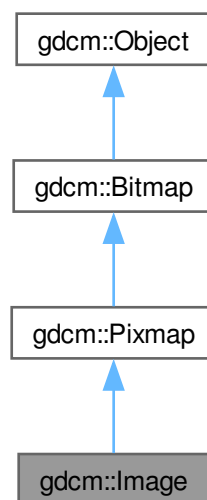
- [gdcmElement.h](#)

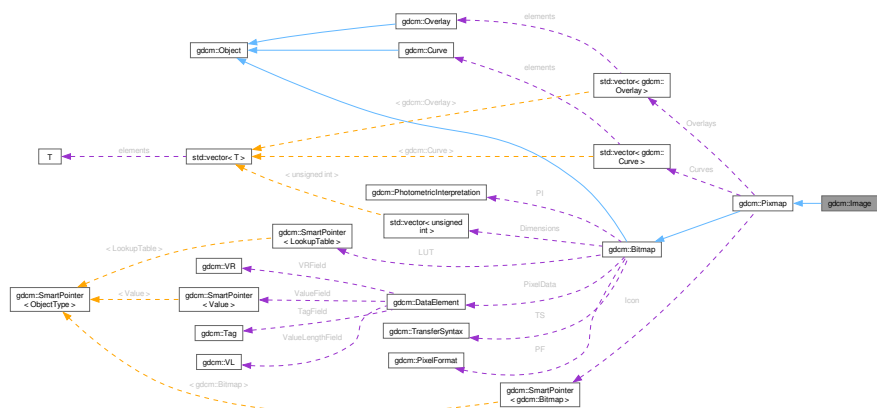
12.146 gdcm::Image Class Reference

[Image](#).

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:





- Image ()

- `Image ()`
 - `~Image ()` override=default
 - `const double * GetDirectionCosines ()` const
 - `double GetDirectionCosines (unsigned int idx)` const
 - `double GetIntercept ()` const
 - `const double * GetOrigin ()` const
 - `double GetOrigin (unsigned int idx)` const
 - `double GetSlope ()` const
 - `const double * GetSpacing ()` const
 - `double GetSpacing (unsigned int idx)` const
 - `void Print (std::ostream &os)` const override
- print
- `void SetDirectionCosines (const double dircos[6])`
 - `void SetDirectionCosines (const float dircos[6])`
 - `void SetDirectionCosines (unsigned int idx, double dircos)`
 - `void SetIntercept (double intercept)`
- intercept
- `void SetOrigin (const double origin[3])`
 - `void SetOrigin (const float origin[3])`
 - `void SetOrigin (unsigned int idx, double ori)`
 - `void SetSlope (double slope)`
- slope
- `void SetSpacing (const double spacing[3])`
 - `void SetSpacing (unsigned int idx, double spacing)`

Public Member Functions inherited from [gdcm::Pixmap](#)

- [Pixmap](#) ()
- [~Pixmap](#) () override
- bool [AreOverlaysInPixelData](#) () const override
returns if Overlays are stored in the unused bit of the pixel data:
- [Curve](#) & [GetCurve](#) (size_t i=0)
[Curve](#): group 50xx.
- const [Curve](#) & [GetCurve](#) (size_t i=0) const
- [IconImage](#) & [GetIconImage](#) ()
- const [IconImage](#) & [GetIconImage](#) () const
Set/Get Icon Image.
- size_t [GetNumberOfCurves](#) () const
- size_t [GetNumberOfOverlays](#) () const
- [Overlay](#) & [GetOverlay](#) (size_t i=0)
[Overlay](#): group 60xx.
- const [Overlay](#) & [GetOverlay](#) (size_t i=0) const
- void [Print](#) (std::ostream &) const override
- void [RemoveOverlay](#) (size_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size_t n)
- void [SetNumberOfOverlays](#) (size_t n)
- bool [UnusedBitsPresentInPixelData](#) () const override
returns if there are unused bits in the pixel data

Public Member Functions inherited from [gdcm::Bitmap](#)

- [Bitmap](#) ()
- [~Bitmap](#) () override
- void [Clear](#) ()
- bool [GetBuffer](#) (char *buffer) const
Access the raw data.
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int * [GetDimensions](#) () const
Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
INTERNAL do not use.
- unsigned int [GetNumberOfDimensions](#) () const
Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
return the photometric interpretation
- [PixelFormat](#) & [GetPixelFormat](#) ()

- const [PixelFormat](#) & [GetPixelFormat](#) () const
Get/Set [PixelFormat](#).
- unsigned int [GetPlanarConfiguration](#) () const
return the planar configuration
- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const
Return whether or not the image was compressed using a lossy compressor or not.
- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)
Specifically set that the image was compressed using a lossy compression mechanism.
- void [SetLUT](#) ([LookupTable](#) const &lut)
Set/Get LUT.
- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)
Transfer syntax.

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Additional Inherited Members

Protected Types inherited from [gdcm::Bitmap](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions inherited from [gdcm::Bitmap](#)

- bool [ComputeLossyFlag](#) ()
- bool [GetBuffer2](#) (std::ostream &os) const
- bool [TryJPEG2000Codec](#) (char *buffer, bool &lossyflag) const
- bool [TryJPEG2000Codec2](#) (std::ostream &os) const
- bool [TryJPEGCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryJPEGCodec2](#) (std::ostream &os) const
- bool [TryJPEGLSCCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryKAKADUCCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryPVRGCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryRAWCodec](#) (char *buffer, bool &lossyflag) const
- bool [TryRLECodec](#) (char *buffer, bool &lossyflag) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::Pixmap](#)

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)

Protected Attributes inherited from [gdcm::Bitmap](#)

- std::vector< unsigned int > [Dimensions](#)
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- [DataElement](#) [PixelData](#)
- unsigned int [PlanarConfiguration](#)
- [TransferSyntax](#) [TS](#)

12.146.1 Detailed Description

[Image](#).

This is the container for an [Image](#) in the general sense. From this container you should be able to request information like:

- Origin
- Dimension
- [PixelFormat](#) ... But also to retrieve the image as a raw buffer (char *) Since we have to deal with both RAW data and JPEG stream (which internally encode all the above information) this API might seem redundant. One way to solve that would be to subclass [Image](#) with [JPEGImage](#) which would from the stream extract the header info and fill it to please [Image](#)...well except origin for instance

Basically you can see it as a storage for the Pixel Data element (7fe0,0010).

Warning

This class does some heuristics to guess the [Spacing](#) but is not compatible with DICOM CP-586. In case of doubt use [PixmapReader](#) instead

See also

[ImageReader](#) [PixmapReader](#)

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcmm.cxx](#).

12.146.2 Constructor & Destructor Documentation

12.146.2.1 [Image\(\)](#)

`gdcmm::Image::Image ()` [inline]

12.146.2.2 [~Image\(\)](#)

`gdcmm::Image::~~Image ()` [override], [default]

12.146.3 Member Function Documentation

12.146.3.1 GetDirectionCosines() [1/2]

```
const double * gdcm::Image::GetDirectionCosines () const
```

Return a 6-tuples specifying the direction cosines A default value of (1,0,0,0,1,0) will be return when the direction cosines was not specified.

12.146.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (  
    unsigned int idx) const
```

12.146.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept () const [inline]
```

12.146.3.4 GetOrigin() [1/2]

```
const double * gdcm::Image::GetOrigin () const
```

Return a 3-tuples specifying the origin Will return (0,0,0) if the origin was not specified.

Examples

[HelloVizWorld.cxx](#).

12.146.3.5 GetOrigin() [2/2]

```
double gdcm::Image::GetOrigin (  
    unsigned int idx) const
```

12.146.3.6 GetSlope()

```
double gdcm::Image::GetSlope () const [inline]
```

12.146.3.7 GetSpacing() [1/2]

```
const double * gdcm::Image::GetSpacing () const
```

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

12.146.3.8 GetSpacing() [2/2]

```
double gdcmm::Image::GetSpacing (  
    unsigned int idx) const
```

12.146.3.9 Print()

```
void gdcmm::Image::Print (  
    std::ostream & os) const    [override], [virtual]
```

print

Reimplemented from [gdcmm::Bitmap](#).

Examples

[CompressImage.cxx](#), and [PatchFile.cxx](#).

12.146.3.10 SetDirectionCosines() [1/3]

```
void gdcmm::Image::SetDirectionCosines (  
    const double dircos[6])
```

12.146.3.11 SetDirectionCosines() [2/3]

```
void gdcmm::Image::SetDirectionCosines (  
    const float dircos[6])
```

12.146.3.12 SetDirectionCosines() [3/3]

```
void gdcmm::Image::SetDirectionCosines (  
    unsigned int idx,  
    double dircos)
```

12.146.3.13 SetIntercept()

```
void gdcmm::Image::SetIntercept (  
    double intercept)    [inline]
```

intercept

Examples

[TemplateEmptyImage.cxx](#).

12.146.3.14 SetOrigin() [1/3]

```
void gdcm::Image::SetOrigin (  
    const double origin[3])
```

12.146.3.15 SetOrigin() [2/3]

```
void gdcm::Image::SetOrigin (  
    const float origin[3])
```

12.146.3.16 SetOrigin() [3/3]

```
void gdcm::Image::SetOrigin (  
    unsigned int idx,  
    double ori)
```

12.146.3.17 SetSlope()

```
void gdcm::Image::SetSlope (  
    double slope) [inline]
```

slope

Examples

[TemplateEmptyImage.cxx](#).

12.146.3.18 SetSpacing() [1/2]

```
void gdcm::Image::SetSpacing (  
    const double spacing[3])
```

Examples

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.146.3.19 SetSpacing() [2/2]

```
void gdcm::Image::SetSpacing (  
    unsigned int idx,  
    double spacing)
```

The documentation for this class was generated from the following file:

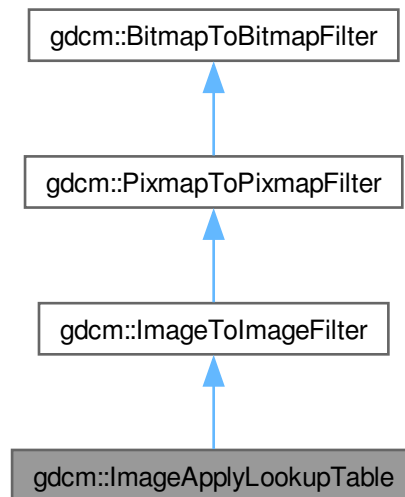
- [gdcmImage.h](#)

12.147 gdcm::ImageApplyLookupTable Class Reference

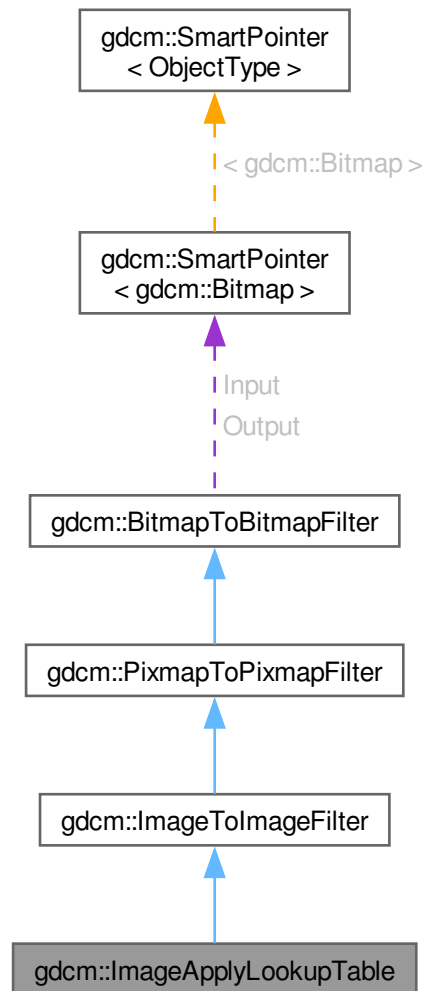
[ImageApplyLookupTable](#) class.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for `gdcm::ImageApplyLookupTable`:



Public Member Functions

- `ImageApplyLookupTable ()`
- `~ImageApplyLookupTable ()`
- `bool Apply ()`
Apply.
- `void SetRGB8 (bool b)`
RGB8 ?

Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.147.1 Detailed Description

[ImageApplyLookupTable](#) class.

It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image

12.147.2 Constructor & Destructor Documentation

12.147.2.1 ImageApplyLookupTable()

[gdcm::ImageApplyLookupTable::ImageApplyLookupTable](#) ()

12.147.2.2 ~ImageApplyLookupTable()

gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ()

12.147.3 Member Function Documentation

12.147.3.1 Apply()

bool gdcm::ImageApplyLookupTable::Apply ()

Apply.

12.147.3.2 SetRGB8()

void gdcm::ImageApplyLookupTable::SetRGB8 (
 bool b)

RGB8 ?

The documentation for this class was generated from the following file:

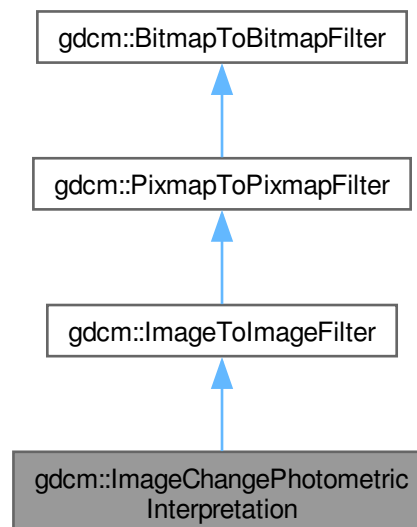
- [gdcmImageApplyLookupTable.h](#)

12.148 gdcm::ImageChangePhotometricInterpretation Class Reference

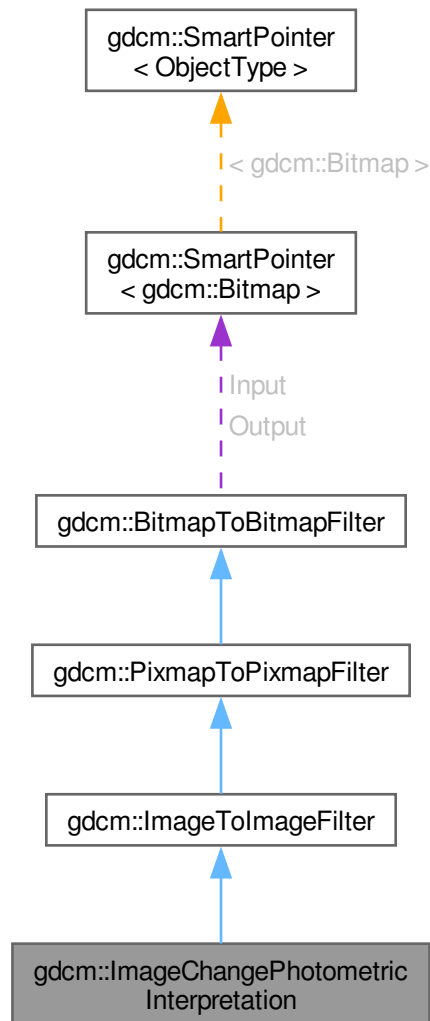
[ImageChangePhotometricInterpretation](#) class.

#include <gdcmImageChangePhotometricInterpretation.h>

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- `ImageChangePhotometricInterpretation ()`
- `~ImageChangePhotometricInterpretation ()=default`
- `bool Change ()`
Change.
- `const PhotometricInterpretation & GetPhotometricInterpretation () const`
- `void SetPhotometricInterpretation (PhotometricInterpretation const &pi)`
Set/Get requested `PhotometricInterpretation`.

Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Static Public Member Functions

- template<typename T>
static void [RGB2YBR](#) (T ybr[3], const T rgb[3], unsigned short storedbits=8)
- template<typename T>
static void [YBR2RGB](#) (T rgb[3], const T ybr[3], unsigned short storedbits=8)

Protected Member Functions

- bool [ChangeMonochrome](#) ()
- bool [ChangeRGB2YBR](#) ()
- bool [ChangeYBR2RGB](#) ()

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.148.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class.

Class to change the Photometric Interpretation of an input DICOM

12.148.2 Constructor & Destructor Documentation

12.148.2.1 ImageChangePhotometricInterpretation()

`gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ()` [inline]

12.148.2.2 ~ImageChangePhotometricInterpretation()

`gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ()` [default]

12.148.3 Member Function Documentation

12.148.3.1 Change()

`bool gdcm::ImageChangePhotometricInterpretation::Change ()`

Change.

References [RGB2YBR\(\)](#), and [YBR2RGB\(\)](#).

12.148.3.2 ChangeMonochrome()

`bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ()` [protected]

12.148.3.3 ChangeRGB2YBR()

`bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR ()` [protected]

12.148.3.4 ChangeYBR2RGB()

`bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB ()` [protected]

12.148.3.5 GetPhotometricInterpretation()

`const PhotometricInterpretation & gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation () const` [inline]

12.148.3.6 RGB2YBR()

```
template<typename T>
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (
    T ybr[3],
    const T rgb[3],
    unsigned short storedbits = 8) [static]
```

colorspace conversion (based on CCIR Recommendation 601-2) -> T.871

References [gdcm::Clamp\(\)](#), [gdcm_assert](#), and [gdcm::Round\(\)](#).

Referenced by [Change\(\)](#).

12.148.3.7 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi) [inline]
```

Set/Get requested [PhotometricInterpretation](#).

12.148.3.8 YBR2RGB()

```
template<typename T>
void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (
    T rgb[3],
    const T ybr[3],
    unsigned short storedbits = 8) [static]
```

References [gdcm::Clamp\(\)](#), [gdcm_assert](#), and [gdcm::Round\(\)](#).

Referenced by [Change\(\)](#).

The documentation for this class was generated from the following file:

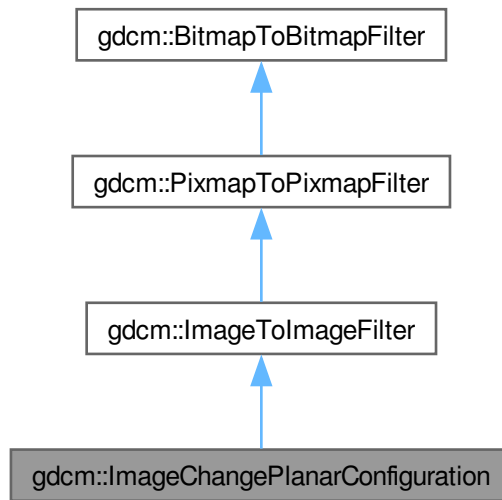
- [gdcmImageChangePhotometricInterpretation.h](#)

12.149 gdcm::ImageChangePlanarConfiguration Class Reference

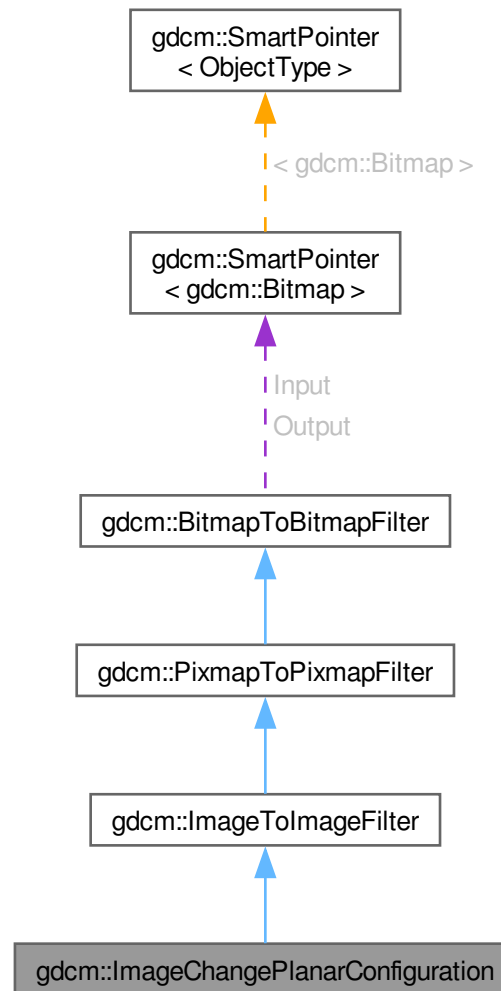
[ImageChangePlanarConfiguration](#) class.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:



Collaboration diagram for `gdcm::ImageChangePlanarConfiguration`:



Public Member Functions

- `ImageChangePlanarConfiguration ()`
- `~ImageChangePlanarConfiguration ()=default`
- `bool Change ()`
Change.
- `unsigned int GetPlanarConfiguration () const`
- `void SetPlanarConfiguration (unsigned int pc)`
Set/Get requested PlanarConfiguration.

Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Static Public Member Functions

- template<typename T>
static size_t [RGBPixelsToRGBPlanes](#) (T *r, T *g, T *b, const T *rgb, size_t s)
- template<typename T>
static size_t [RGBPlanesToRGBPixels](#) (T *out, const T *r, const T *g, const T *b, size_t s)

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.149.1 Detailed Description

[ImageChangePlanarConfiguration](#) class.

Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0

12.149.2 Constructor & Destructor Documentation

12.149.2.1 ImageChangePlanarConfiguration()

`gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ()` [inline]

12.149.2.2 ~ImageChangePlanarConfiguration()

`gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ()` [default]

12.149.3 Member Function Documentation

12.149.3.1 Change()

`bool gdcm::ImageChangePlanarConfiguration::Change ()`

Change.

12.149.3.2 GetPlanarConfiguration()

`unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration () const` [inline]

12.149.3.3 RGBPixelsToRGBPlanes()

```
template<typename T>
size_t gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes (
    T * r,
    T * g,
    T * b,
    const T * rgb,
    size_t s) [static]
```

Convert a regular RGB pixel image (R,G,B,R,G,B...) into a planar R,G,B image (R,R...,G,G...B,B)

Warning

this works on a frame basis, you need to loop over all frames in multiple frames image to apply this function

References [gdcm_assert](#).

12.149.3.4 RGBPlanesToRGBPixels()

```
template<typename T>
size_t gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels (
    T * out,
    const T * r,
    const T * g,
    const T * b,
    size_t s) [static]
```

s is the size of one plane (r,g or b). Thus the output buffer needs to be at least 3*s bytes long s can be seen as the number of RGB pixels in the output

References [gdcm_assert](#).

12.149.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
    unsigned int pc) [inline]
```

Set/Get requested PlanarConfiguration.

The documentation for this class was generated from the following file:

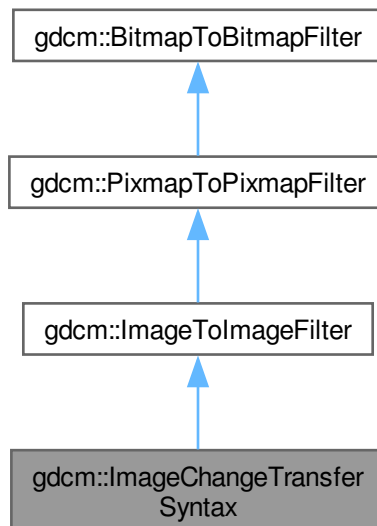
- [gdcmImageChangePlanarConfiguration.h](#)

12.150 gdcm::ImageChangeTransferSyntax Class Reference

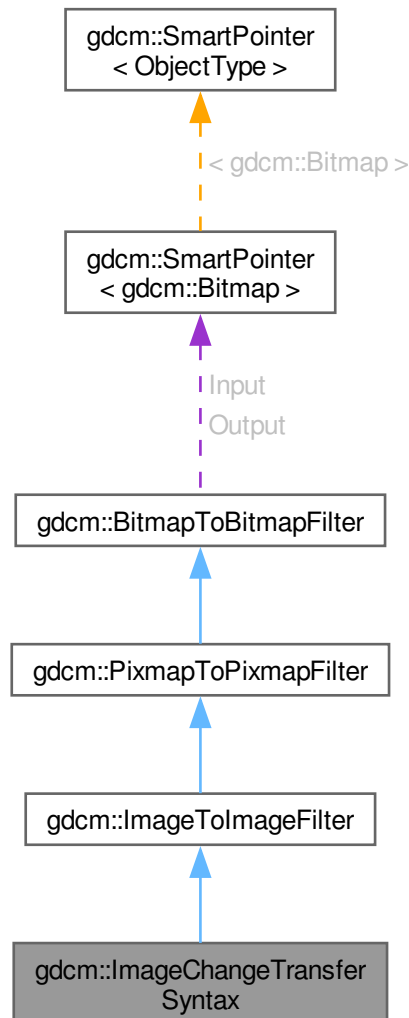
[ImageChangeTransferSyntax](#) class.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for `gdcm::ImageChangeTransferSyntax`:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- `ImageChangeTransferSyntax ()`
- `~ImageChangeTransferSyntax ()=default`
- `bool Change ()`
Change.
- `const TransferSyntax & GetTransferSyntax () const`
Get Transfer Syntax.
- `void SetCompressIconImage (bool b)`
- `void SetForce (bool f)`

- void [SetTransferSyntax](#) (const [TransferSyntax](#) &ts)
Set target Transfer Syntax.
- void [SetUserCodec](#) ([ImageCodec](#) *ic)

Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Protected Member Functions

- bool [TryJPEG2000Codec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGLSCCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryRAWCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryRLECodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.150.1 Detailed Description

[ImageChangeTransferSyntax](#) class.

Class to change the transfer syntax of an input DICOM

If only Force param is set but no input [TransferSyntax](#) is set, it is assumed that user only wants to inspect encapsulated stream (advanced dev. option).

When using UserCodec it is very important that the [TransferSyntax](#) (as set in SetTransferSyntax) is actually understood by UserCodec (ie. UserCodec->CanCode(TransferSyntax)). Otherwise the behavior is to use a default codec.

See also

[JPEGCodec](#) [JPEGLSCodec](#) [JPEG2000Codec](#)

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ExplicitLittleEndian.cs](#), and [StandardizeFiles.cs](#).

12.150.2 Constructor & Destructor Documentation

12.150.2.1 ImageChangeTransferSyntax()

gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax () [inline]

12.150.2.2 ~ImageChangeTransferSyntax()

gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax () [default]

12.150.3 Member Function Documentation

12.150.3.1 Change()

bool gdcm::ImageChangeTransferSyntax::Change ()

Change.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ExplicitLittleEndian.cs](#), and [StandardizeFiles.cs](#).

12.150.3.2 GetTransferSyntax()

```
const TransferSyntax & gdcm::ImageChangeTransferSyntax::GetTransferSyntax () const [inline]
```

Get Transfer Syntax.

12.150.3.3 SetCompressIconImage()

```
void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (  
    bool b) [inline]
```

Decide whether or not to also compress the Icon [Image](#) using the same Transfer Syntax. Default is to simply decompress icon image

Examples

[StandardizeFiles.cs](#).

12.150.3.4 SetForce()

```
void gdcm::ImageChangeTransferSyntax::SetForce (  
    bool f) [inline]
```

When target Transfer Syntax is identical to input target syntax, no operation is actually done. This is an issue when someone wants to re-compress using GDCM internal implementation a JPEG (for example) image

Examples

[ExplicitLittleEndian.cs](#), and [StandardizeFiles.cs](#).

12.150.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (  
    const TransferSyntax & ts) [inline]
```

Set target Transfer Syntax.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ExplicitLittleEndian.cs](#), and [StandardizeFiles.cs](#).

12.150.3.6 SetUserCodec()

```
void gdcmm::ImageChangeTransferSyntax::SetUserCodec (
    ImageCodec * ic) [inline]
```

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

if the codec 'ic' is not compatible with the [TransferSyntax](#) requested, it will not be used. It is the user responsibility to check that UserCodec->CanCode(TransferSyntax)

Examples

[CompressLossyJPEG.cs](#).

12.150.3.7 TryJPEG2000Codec()

```
bool gdcmm::ImageChangeTransferSyntax::TryJPEG2000Codec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output) [protected]
```

12.150.3.8 TryJPEGCodec()

```
bool gdcmm::ImageChangeTransferSyntax::TryJPEGCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output) [protected]
```

12.150.3.9 TryJPEGLSCodec()

```
bool gdcmm::ImageChangeTransferSyntax::TryJPEGLSCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output) [protected]
```

12.150.3.10 TryRAWCodec()

```
bool gdcmm::ImageChangeTransferSyntax::TryRAWCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output) [protected]
```

12.150.3.11 TryRLECodec()

```
bool gdcM::ImageChangeTransferSyntax::TryRLECodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output) [protected]
```

The documentation for this class was generated from the following file:

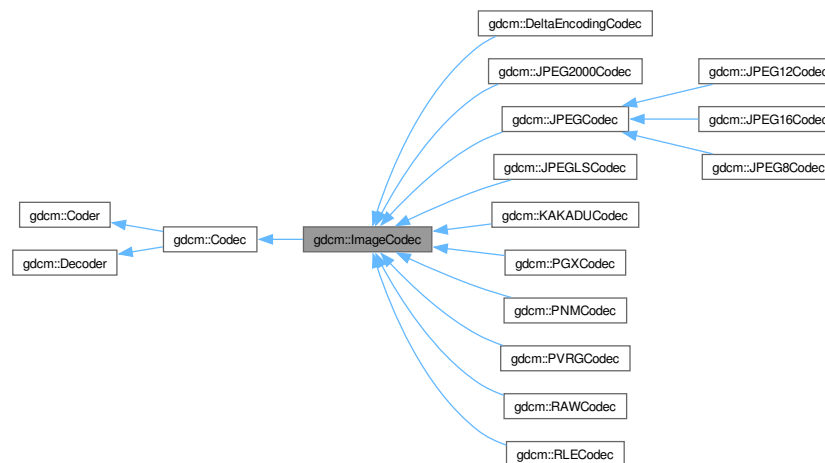
- [gdcMImageChangeTransferSyntax.h](#)

12.151 gdcM::ImageCodec Class Reference

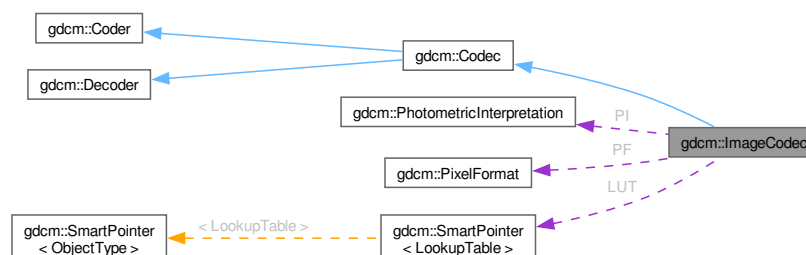
[ImageCodec](#).

```
#include <gdcMImageCodec.h>
```

Inheritance diagram for gdcM::ImageCodec:



Collaboration diagram for gdcM::ImageCodec:



Public Member Functions

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override
Return whether this decoder support this transfer syntax (can decode it).
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- virtual [ImageCodec](#) * [Clone](#) () const =0
- bool [Decode](#) ([DataElement](#) const &is_, [DataElement](#) &os) override
Decode.
- const unsigned int * [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Types

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

Friends

- class [FileChangeTransferSyntax](#)
- class [ImageChangePhotometricInterpretation](#)

12.151.1 Detailed Description

[ImageCodec](#).

Note

Main codec, this is a central place for all implementation

Examples

[FileChangeTSLossy.cs](#).

12.151.2 Member Typedef Documentation

12.151.2.1 LUTPtr

typedef [SmartPointer](#)<[LookupTable](#)> [gdcm::ImageCodec::LUTPtr](#) [protected]

12.151.3 Constructor & Destructor Documentation

12.151.3.1 ImageCodec()

[gdcm::ImageCodec::ImageCodec](#) ()

Referenced by [Clone\(\)](#), [gdcm::JPEG2000Codec::Clone\(\)](#), [gdcm::JPEGCodec::Clone\(\)](#), [gdcm::JPEGLSCodec::Clone\(\)](#), [gdcm::KAKADUCodec::Clone\(\)](#), [gdcm::PGXCodec::Clone\(\)](#), [gdcm::PNMCodec::Clone\(\)](#), [gdcm::PVRGCodec::Clone\(\)](#), [gdcm::RAWCodec::Clone\(\)](#), and [gdcm::RLECodec::Clone\(\)](#).

12.151.3.2 ~ImageCodec()

[gdcm::ImageCodec::~~ImageCodec](#) () [override]

12.151.4 Member Function Documentation

12.151.4.1 AppendFrameEncode()

virtual bool [gdcm::ImageCodec::AppendFrameEncode](#) (
 std::ostream & out,
 const char * data,
 size_t datalen) [protected], [virtual]

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), and [gdcm::RLECodec](#).

12.151.4.2 AppendRowEncode()

```
virtual bool gdcM::ImageCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen) [protected], [virtual]
```

Reimplemented in [gdcM::JPEG2000Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), and [gdcM::RLECodec](#).

12.151.4.3 CanCode()

```
bool gdcM::ImageCodec::CanCode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implements [gdcM::Coder](#).

Reimplemented in [gdcM::JPEG2000Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::KAKADUCodec](#), [gdcM::PGXCodec](#), [gdcM::PNMCodec](#), [gdcM::PVRGCodec](#), [gdcM::RAWCodec](#), and [gdcM::RLECodec](#).

12.151.4.4 CanDecode()

```
bool gdcM::ImageCodec::CanDecode (
    TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implements [gdcM::Decoder](#).

Reimplemented in [gdcM::JPEG2000Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::KAKADUCodec](#), [gdcM::PGXCodec](#), [gdcM::PNMCodec](#), [gdcM::PVRGCodec](#), [gdcM::RAWCodec](#), and [gdcM::RLECodec](#).

12.151.4.5 CleanupUnusedBits()

```
bool gdcM::ImageCodec::CleanupUnusedBits (
    char * data,
    size_t datalen)
```

12.151.4.6 Clone()

```
virtual ImageCodec * gdcM::ImageCodec::Clone () const [pure virtual]
```

Implemented in [gdcM::JPEG2000Codec](#), [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::KAKADUCodec](#), [gdcM::PGXCodec](#), [gdcM::PNMCodec](#), [gdcM::PVRGCodec](#), [gdcM::RAWCodec](#), and [gdcM::RLECodec](#).

References [ImageCodec\(\)](#).

12.151.4.7 Decode()

```
bool gdcmm::ImageCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcmm::Decoder](#).

Reimplemented in [gdcmm::JPEG2000Codec](#), [gdcmm::JPEGCodec](#), [gdcmm::JPEGLSCCodec](#), [gdcmm::KAKADUCodec](#), [gdcmm::PVRGCodec](#), [gdcmm::RAWCodec](#), and [gdcmm::RLECodec](#).

12.151.4.8 DecodeByStreams()

```
bool gdcmm::ImageCodec::DecodeByStreams (
    std::istream & is_,
    std::ostream & os) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::Decoder](#).

Reimplemented in [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::JPEG8Codec](#), [gdcmm::JPEGCodec](#), [gdcmm::RAWCodec](#), and [gdcmm::RLECodec](#).

12.151.4.9 DoByteSwap()

```
bool gdcmm::ImageCodec::DoByteSwap (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.10 DoInvertMonochrome()

```
bool gdcmm::ImageCodec::DoInvertMonochrome (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.11 DoOverlayCleanup()

```
bool gdcmm::ImageCodec::DoOverlayCleanup (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.12 DoPaddedCompositePixelCode()

```
bool gdcmm::ImageCodec::DoPaddedCompositePixelCode (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.13 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.14 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.15 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.16 DoYBRFull422()

```
bool gdcm::ImageCodec::DoYBRFull422 (
    std::istream & is_,
    std::ostream & os) [protected]
```

12.151.4.17 GetDimensions()

```
const unsigned int * gdcm::ImageCodec::GetDimensions () const [inline]
```

References [Dimensions](#).

12.151.4.18 GetHeaderInfo()

```
virtual bool gdcm::ImageCodec::GetHeaderInfo (
    std::istream & is_,
    TransferSyntax & ts) [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG8Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::PGXCodec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), and [gdcm::RLECodec](#).

12.151.4.19 GetLossyFlag()

```
bool gdcm::ImageCodec::GetLossyFlag () const
```

12.151.4.20 GetLUT()

const [LookupTable](#) & gdcm::ImageCodec::GetLUT () const [inline]

References [LUT](#).

12.151.4.21 GetNeedByteSwap()

bool gdcm::ImageCodec::GetNeedByteSwap () const [inline]

References [NeedByteSwap](#).

12.151.4.22 GetNumberOfDimensions()

unsigned int gdcm::ImageCodec::GetNumberOfDimensions () const

12.151.4.23 GetPhotometricInterpretation()

const [PhotometricInterpretation](#) & gdcm::ImageCodec::GetPhotometricInterpretation () const

12.151.4.24 GetPixelFormat() [1/2]

[PixelFormat](#) & gdcm::ImageCodec::GetPixelFormat () [inline]

Examples

[GetJPEGSamplePrecision.cxx](#).

References [PF](#).

12.151.4.25 GetPixelFormat() [2/2]

const [PixelFormat](#) & gdcm::ImageCodec::GetPixelFormat () const [inline]

References [PF](#).

12.151.4.26 GetPlanarConfiguration()

unsigned int gdcm::ImageCodec::GetPlanarConfiguration () const [inline]

References [PlanarConfiguration](#).

12.151.4.27 IsFrameEncoder()

virtual bool gdcm::ImageCodec::IsFrameEncoder () [protected], [virtual]

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCCodec](#), and [gdcm::RLECodec](#).

12.151.4.28 IsLossy()

bool gdcm::ImageCodec::IsLossy () const

12.151.4.29 IsRowEncoder()

virtual bool gdcm::ImageCodec::IsRowEncoder () [protected], [virtual]

Reimplemented in [gdcm::JPEG2000Codec](#), [gdcm::JPEGCodec](#), [gdcm::JPEGLSCCodec](#), and [gdcm::RLECodec](#).

12.151.4.30 IsValid()

virtual bool gdcm::ImageCodec::IsValid (
 [PhotometricInterpretation](#) const & pi) [protected], [virtual]

Reimplemented in [gdcm::JPEGCodec](#).

12.151.4.31 SetDimensions() [1/2]

void gdcm::ImageCodec::SetDimensions (
 const std::vector< unsigned int > & d)

12.151.4.32 SetDimensions() [2/2]

void gdcm::ImageCodec::SetDimensions (
 const unsigned int d[3])

Examples

[ExtractIconFromFile.cxx](#).

12.151.4.33 SetLossyFlag()

void gdcm::ImageCodec::SetLossyFlag (
 bool l)

12.151.4.34 SetLUT()

```
void gdcmm::ImageCodec::SetLUT (  
    LookupTable const & lut) [inline]
```

Examples

[ExtractIconFromFile.cxx](#).

References [LUT](#).

12.151.4.35 SetNeedByteSwap()

```
void gdcmm::ImageCodec::SetNeedByteSwap (  
    bool b) [inline]
```

References [NeedByteSwap](#).

12.151.4.36 SetNeedOverlayCleanup()

```
void gdcmm::ImageCodec::SetNeedOverlayCleanup (  
    bool b) [inline]
```

References [NeedOverlayCleanup](#).

12.151.4.37 SetNumberOfDimensions()

```
void gdcmm::ImageCodec::SetNumberOfDimensions (  
    unsigned int dim)
```

12.151.4.38 SetPhotometricInterpretation()

```
void gdcmm::ImageCodec::SetPhotometricInterpretation (  
    PhotometricInterpretation const & pi)
```

Examples

[ExtractIconFromFile.cxx](#).

12.151.4.39 SetPixelFormat()

virtual void gdcmm::ImageCodec::SetPixelFormat (
 PixelFormat const & pf) [inline], [virtual]

Reimplemented in [gdcmm::JPEGCodec](#).

Examples

[ExtractIconFromFile.cxx](#).

References [PF](#).

12.151.4.40 SetPlanarConfiguration()

void gdcmm::ImageCodec::SetPlanarConfiguration (
 unsigned int pc) [inline]

References [gdcmm_assert](#), and [PlanarConfiguration](#).

12.151.4.41 StartEncode()

virtual bool gdcmm::ImageCodec::StartEncode (
 std::ostream & os) [protected], [virtual]

Reimplemented in [gdcmm::JPEG2000Codec](#), [gdcmm::JPEGCodec](#), [gdcmm::JPEGLSCCodec](#), and [gdcmm::RLECodec](#).

12.151.4.42 StopEncode()

virtual bool gdcmm::ImageCodec::StopEncode (
 std::ostream & os) [protected], [virtual]

Reimplemented in [gdcmm::JPEG2000Codec](#), [gdcmm::JPEGCodec](#), [gdcmm::JPEGLSCCodec](#), and [gdcmm::RLECodec](#).

12.151.5 Friends And Related Symbol Documentation

12.151.5.1 FileChangeTransferSyntax

friend class FileChangeTransferSyntax [friend]

This is a high level API to encode in a streaming fashion. Each plugin will handle differently the caching mechanism so that a limited memory is used when compressing dataset. [Codec](#) will fall into two categories:

- Full row encoder: only a single scanline (row) of data is needed to be loaded at a time;
- Full frame encoder (default): a complete frame (row x col) is needed to be loaded at a time

References [FileChangeTransferSyntax](#).

Referenced by [FileChangeTransferSyntax](#).

12.151.5.2 ImageChangePhotometricInterpretation

friend class ImageChangePhotometricInterpretation [friend]

References [ImageChangePhotometricInterpretation](#).

Referenced by [ImageChangePhotometricInterpretation](#).

12.151.6 Member Data Documentation

12.151.6.1 Dimensions

unsigned int gdcm::ImageCodec::Dimensions[3] [protected]

Referenced by [GetDimensions\(\)](#).

12.151.6.2 LossyFlag

bool gdcm::ImageCodec::LossyFlag [protected]

12.151.6.3 LUT

[LUTPtr](#) gdcm::ImageCodec::LUT [protected]

Referenced by [GetLUT\(\)](#), and [SetLUT\(\)](#).

12.151.6.4 NeedByteSwap

bool gdcm::ImageCodec::NeedByteSwap [protected]

Referenced by [GetNeedByteSwap\(\)](#), and [SetNeedByteSwap\(\)](#).

12.151.6.5 NeedOverlayCleanup

bool gdcm::ImageCodec::NeedOverlayCleanup [protected]

Referenced by [SetNeedOverlayCleanup\(\)](#).

12.151.6.6 NumberOfDimensions

unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]

12.151.6.7 PF

[PixelFormat](#) gdcm::ImageCodec::PF [protected]

Referenced by [GetPixelFormat\(\)](#), [GetPixelFormat\(\)](#), and [SetPixelFormat\(\)](#).

12.151.6.8 PI

[PhotometricInterpretation](#) gdcm::ImageCodec::PI [protected]

12.151.6.9 PlanarConfiguration

unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]

Referenced by [GetPlanarConfiguration\(\)](#), and [SetPlanarConfiguration\(\)](#).

12.151.6.10 RequestPaddedCompositePixelCode

bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]

12.151.6.11 RequestPlanarConfiguration

bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]

The documentation for this class was generated from the following file:

- [gdcmImageCodec.h](#)

12.152 gdcm::ImageConverter Class Reference

[Image](#) Converter.

```
#include <gdcmImageConverter.h>
```

Public Member Functions

- [ImageConverter](#) ()
- [~ImageConverter](#) ()
- void [Convert](#) ()
- const [Image](#) & [GetOutput](#) () const
- void [SetInput](#) ([Image](#) const &input)

12.152.1 Detailed Description

[Image](#) Converter.

Note

This is the class used to convert from one [Image](#) to another. This is typically used to convert let say YBR JPEG compressed [Image](#) to a RAW RGB [Image](#). So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

12.152.2 Constructor & Destructor Documentation

12.152.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ()
```

12.152.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~~ImageConverter ()
```

12.152.3 Member Function Documentation

12.152.3.1 Convert()

```
void gdcm::ImageConverter::Convert ()
```

12.152.3.2 GetOutput()

```
const Image & gdcm::ImageConverter::GetOutput () const
```

12.152.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (  
    Image const & input)
```

The documentation for this class was generated from the following file:

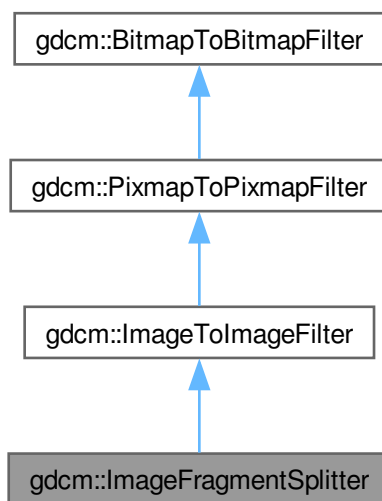
- [gdcmImageConverter.h](#)

12.153 gdcm::ImageFragmentSplitter Class Reference

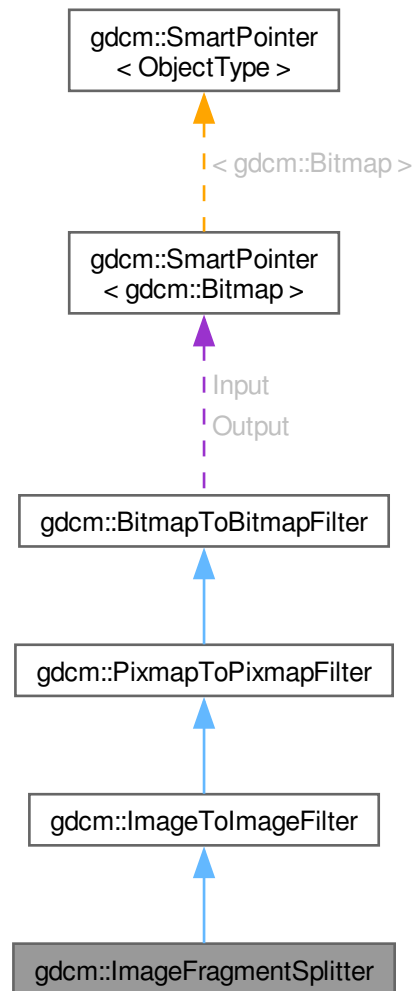
[ImageFragmentSplitter](#) class.

```
#include <gdcmImageFragmentSplitter.h>
```

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- `ImageFragmentSplitter ()`
- `~ImageFragmentSplitter ()=default`
- `unsigned int GetFragmentSizeMax () const`
- `void SetForce (bool f)`
- `void SetFragmentSizeMax (unsigned int fragsize)`
FragmentSizeMax needs to be an even number.
- `bool Split ()`
Split.

Public Member Functions inherited from [gdcm::ImageToImageFilter](#)

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.153.1 Detailed Description

[ImageFragmentSplitter](#) class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

12.153.2 Constructor & Destructor Documentation

12.153.2.1 ImageFragmentSplitter()

[gdcm::ImageFragmentSplitter::ImageFragmentSplitter](#) () [inline]

12.153.2.2 ~ImageFragmentSplitter()

gdcm::ImageFragmentSplitter::~ImageFragmentSplitter () [default]

12.153.3 Member Function Documentation

12.153.3.1 GetFragmentSizeMax()

unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax () const [inline]

12.153.3.2 SetForce()

void gdcm::ImageFragmentSplitter::SetForce (
bool f) [inline]

When file already has all it's segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

12.153.3.3 SetFragmentSizeMax()

void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (
unsigned int fragsize)

FragmentSizeMax needs to be an even number.

12.153.3.4 Split()

bool gdcm::ImageFragmentSplitter::Split ()

Split.

The documentation for this class was generated from the following file:

- [gdcmImageFragmentSplitter.h](#)

12.154 gdcm::ImageHelper Class Reference

[ImageHelper](#) (internal class, not intended for user level).

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- static [MediaStorage ComputeMediaStorageFromModality](#) (const char *modality, unsigned int dimension=2, [PixelFormat](#) const &pf=[PixelFormat\(\)](#), [PhotometricInterpretation](#) const &pi=[PhotometricInterpretation\(\)](#), double rescaleintercept=0, double rescaleslope=1)
 Moved from [MediaStorage](#) here, since we need extra info stored in [PixelFormat](#) & [PhotometricInterpretation](#).
- static bool [ComputeSpacingFromImagePositionPatient](#) (const std::vector< double > &imageposition, std::vector< double > &spacing)
 DO NOT USE.
- static std::vector< unsigned int > [GetDimensionsValue](#) (const [File](#) &f)
- static bool [GetDirectionCosinesFromDataSet](#) ([DataSet](#) const &ds, std::vector< double > &dircos)
- static std::vector< double > [GetDirectionCosinesValue](#) ([File](#) const &f)
- static bool [GetForcePixelSpacing](#) ()
- static bool [GetForceRescaleInterceptSlope](#) ()
- static [SmartPointer< LookupTable >](#) [GetLUT](#) ([File](#) const &f)
 returns the lookup table of an image file
- static std::vector< double > [GetOriginValue](#) ([File](#) const &f)
 Set/Get Origin (IPP) from/to a file.
- static [PhotometricInterpretation](#) [GetPhotometricInterpretationValue](#) ([File](#) const &f)
- static [PixelFormat](#) [GetPixelFormatValue](#) (const [File](#) &f)
- static unsigned int [GetPlanarConfigurationValue](#) (const [File](#) &f)
- static bool [GetPMSRescaleInterceptSlope](#) ()
- static const [ByteValue](#) * [GetPointerFromElement](#) ([Tag](#) const &tag, [File](#) const &f)
- static bool [GetRealWorldValueMappingContent](#) ([File](#) const &f, [RealWorldValueMappingContent](#) &rwvmc)
- static std::vector< double > [GetRescaleInterceptSlopeValue](#) ([File](#) const &f)
- static bool [GetSecondaryCaptureImagePlaneModule](#) ()
- static std::vector< double > [GetSpacingValue](#) ([File](#) const &f)
 Set/Get [Spacing](#) from/to a [File](#).
- static void [SetDimensionsValue](#) ([File](#) &f, const [Pixmap](#) &img)
- static void [SetDirectionCosinesValue](#) ([DataSet](#) &ds, const std::vector< double > &dircos)
- static void [SetForcePixelSpacing](#) (bool)
- static void [SetForceRescaleInterceptSlope](#) (bool)
- static void [SetOriginValue](#) ([DataSet](#) &ds, const [Image](#) &img)
- static void [SetPMSRescaleInterceptSlope](#) (bool)
- static void [SetRescaleInterceptSlopeValue](#) ([File](#) &f, const [Image](#) &img)
- static void [SetSecondaryCaptureImagePlaneModule](#) (bool)
- static void [SetSpacingValue](#) ([DataSet](#) &ds, const std::vector< double > &spacing)

Static Protected Member Functions

- static [Tag](#) [GetSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)
- static [Tag](#) [GetZSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)

12.154.1 Detailed Description

[ImageHelper](#) (internal class, not intended for user level).

Helper for writing World images in DICOM. DICOM has a 'template' approach to image where MR [Image](#) Storage are distinct object from Enhanced MR [Image](#) Storage. For example the Pixel [Spacing](#) in one object is not at the same position (ie [Tag](#)) as in the other this class is the central (read: fragile) place where all the dispatching is done from a unified view of a world image (typically VTK or ITK point of view) down to the low level DICOM point of view.

Warning

: do not expect the API of this class to be maintained at any point, since as Modalities are added the API might have to be augmented or behavior changed to cope with new modalities.

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [ExtractOneFrame.cs](#).

12.154.2 Member Function Documentation

12.154.2.1 ComputeMediaStorageFromModality()

```
MediaStorage gdcm::ImageHelper::ComputeMediaStorageFromModality (
    const char * modality,
    unsigned int dimension = 2,
    PixelFormat const & pf = PixelFormat(),
    PhotometricInterpretation const & pi = PhotometricInterpretation(),
    double rescaleintercept = 0,
    double rescaleslope = 1) [static]
```

Moved from [MediaStorage](#) here, since we need extra info stored in [PixelFormat](#) & [PhotometricInterpretation](#).

12.154.2.2 ComputeSpacingFromImagePositionPatient()

```
bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
    const std::vector< double > & imageposition,
    std::vector< double > & spacing) [static]
```

DO NOT USE.

12.154.2.3 GetDimensionsValue()

```
std::vector< unsigned int > gdcm::ImageHelper::GetDimensionsValue (
    const File & f) [static]
```

This function checks tags (0x0028, 0x0010) and (0x0028, 0x0011) for the rows and columns of the image in pixels (as opposed to actual distances). The output is {col , row}

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.154.2.4 GetDirectionCosinesFromDataSet()

```
bool gdcM::ImageHelper::GetDirectionCosinesFromDataSet (
    DataSet const & ds,
    std::vector< double > & dircos) [static]
```

12.154.2.5 GetDirectionCosinesValue()

```
std::vector< double > gdcM::ImageHelper::GetDirectionCosinesValue (
    File const & f) [static]
```

Get Direction Cosines (IOP) from/to a file Requires a file because mediastorage must be known

12.154.2.6 GetForcePixelSpacing()

```
bool gdcM::ImageHelper::GetForcePixelSpacing () [static]
```

12.154.2.7 GetForceRescaleInterceptSlope()

```
bool gdcM::ImageHelper::GetForceRescaleInterceptSlope () [static]
```

12.154.2.8 GetLUT()

```
SmartPointer< LookupTable > gdcM::ImageHelper::GetLUT (
    File const & f) [static]
```

returns the lookup table of an image file

12.154.2.9 GetOriginValue()

```
std::vector< double > gdcM::ImageHelper::GetOriginValue (
    File const & f) [static]
```

Set/Get Origin (IPP) from/to a file.

12.154.2.10 GetPhotometricInterpretationValue()

```
PhotometricInterpretation gdcM::ImageHelper::GetPhotometricInterpretationValue (
    File const & f) [static]
```

Examples

[ExtractImageRegion.cs](#).

12.154.2.11 GetPixelFormatValue()

PixelFormat gdcm::ImageHelper::GetPixelFormatValue (
 const **File** & f) [static]

This function returns pixel information about an image from its dataset That includes samples per pixel and bit depth (in that order)

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [ExtractOneFrame.cs](#).

12.154.2.12 GetPlanarConfigurationValue()

unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
 const **File** & f) [static]

12.154.2.13 GetPMSRescaleInterceptSlope()

bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope () [static]

12.154.2.14 GetPointerFromElement()

const **ByteValue** * gdcm::ImageHelper::GetPointerFromElement (
Tag const & tag,
File const & f) [static]

12.154.2.15 GetRealWorldValueMappingContent()

bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
File const & f,
RealWorldValueMappingContent & rwvmc) [static]

12.154.2.16 GetRescaleInterceptSlopeValue()

std::vector< double > gdcm::ImageHelper::GetRescaleInterceptSlopeValue (
File const & f) [static]

Set/Get shift/scale from/to a file

Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage Can't take a dataset because the mediastorage of the file must be known

12.154.2.17 GetSecondaryCaptureImagePlaneModule()

```
bool gdcm::ImageHelper::GetSecondaryCaptureImagePlaneModule () [static]
```

12.154.2.18 GetSpacingTagFromMediaStorage()

```
Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
    MediaStorage const & ms) [static], [protected]
```

12.154.2.19 GetSpacingValue()

```
std::vector< double > gdcm::ImageHelper::GetSpacingValue (
    File const & f) [static]
```

Set/Get [Spacing](#) from/to a [File](#).

12.154.2.20 GetZSpacingTagFromMediaStorage()

```
Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
    MediaStorage const & ms) [static], [protected]
```

12.154.2.21 SetDimensionsValue()

```
void gdcm::ImageHelper::SetDimensionsValue (
    File & f,
    const Pixmap & img) [static]
```

12.154.2.22 SetDirectionCosinesValue()

```
void gdcm::ImageHelper::SetDirectionCosinesValue (
    DataSet & ds,
    const std::vector< double > & dircos) [static]
```

Set Direction Cosines (IOP) from/to a file When [IOD](#) does not defines what is IOP (eg. typically Secondary Capture [Image](#) Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

12.154.2.23 SetForcePixelSpacing()

```
void gdcm::ImageHelper::SetForcePixelSpacing (
    bool ) [static]
```

GDCM 1.x compatibility issue: When using ReWrite an MR [Image](#) Storage would be rewritten as Secondary Capture [Object](#) while still having a Pixel [Spacing](#) tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel [Spacing](#) of the [Image](#)

12.154.2.24 SetForceRescaleInterceptSlope()

```
void gdcm::ImageHelper::SetForceRescaleInterceptSlope (
    bool ) [static]
```

GDCM 1.x compatibility issue: Do not use anymore. This hack was used for some MR [Image](#) Storage generated by Philips Modality. When "Combine MR Rescaling" is set to TRUE, rescaling is removed. But when set to FALSE, the Modality LUT was exported. Internally GDCM now handles this gracefully.

12.154.2.25 SetOriginValue()

```
void gdcm::ImageHelper::SetOriginValue (
    DataSet & ds,
    const Image & img) [static]
```

12.154.2.26 SetPMSRescaleInterceptSlope()

```
void gdcm::ImageHelper::SetPMSRescaleInterceptSlope (
    bool ) [static]
```

Since GDCM 2.6.1 Philips Medical [System](#) are read using the Private Field For Rescale Slope/Intercept by default. This mechanism can be deactivated using the following API: This option has no effect when Force↔RescaleInterceptSlope is set to true GDCM will only read those private attribute but never write them out.

12.154.2.27 SetRescaleInterceptSlopeValue()

```
void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
    File & f,
    const Image & img) [static]
```

12.154.2.28 SetSecondaryCaptureImagePlaneModule()

```
void gdcm::ImageHelper::SetSecondaryCaptureImagePlaneModule (
    bool ) [static]
```

Opt into [Image](#) Plane [Module](#) for Secondary Capture [Image](#) Storage Enable reading [Image](#) Position [Patient](#) (IPP), [Image Orientation Patient](#) (IOP) and Pixel [Spacing](#) (0028,0030) This is a custom extension for some existing dataset (academic)

12.154.2.29 SetSpacingValue()

```
void gdcM::ImageHelper::SetSpacingValue (  
    DataSet & ds,  
    const std::vector< double > & spacing) [static]
```

Warning

You need to call SetSpacingValue after SetOriginValue / SetDirectionCosinesValue

The documentation for this class was generated from the following file:

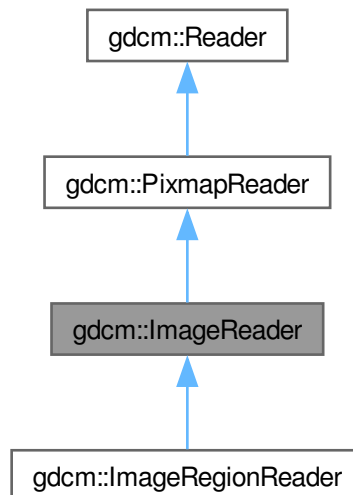
- [gdcMImageHelper.h](#)

12.155 gdcM::ImageReader Class Reference

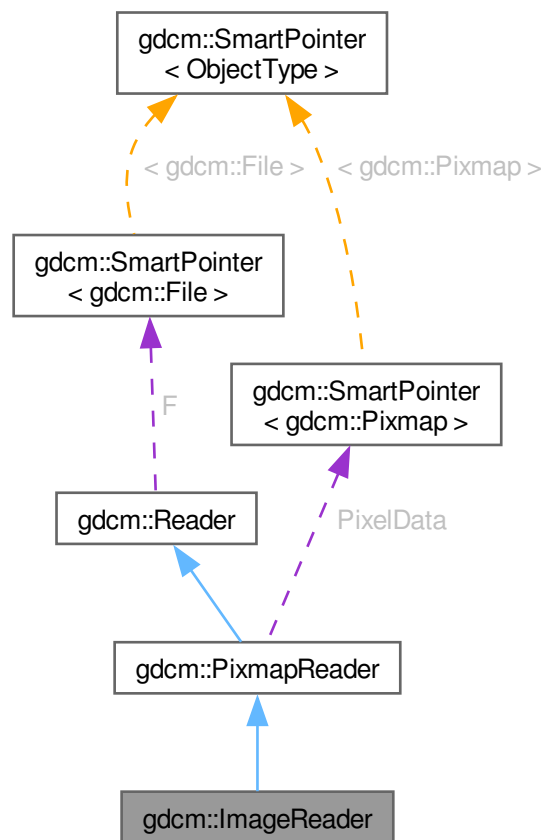
[ImageReader](#).

```
#include <gdcMImageReader.h>
```

Inheritance diagram for gdcM::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- [ImageReader](#) ()
- [~ImageReader](#) () override
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const
Return the read image.
- bool [Read](#) () override

Public Member Functions inherited from [gdcm::PixmapReader](#)

- [PixmapReader](#) ()
- [~PixmapReader](#) () override
- [Pixmap](#) & [GetPixmap](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
Return the read image (need to call [Read\(\)](#) first).
- bool [Read](#) () override

Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()
 - Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
 - Set/Get [File](#).
- size_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
 - Will only read the specified selected private tags.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
 - Will only read the specified selected tags.
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)
 - Set/Get [File](#).
- void [SetFileName](#) (const char *filename__native)
- void [SetStream](#) (std::istream &input__stream)
 - Set the open-ed stream directly.

Protected Member Functions

- bool [ReadACRNEMAIImage](#) () override
- bool [ReadImage](#) ([MediaStorage](#) const &ms) override

Protected Member Functions inherited from [gdcm::PixmapReader](#)

- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream * [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

Additional Inherited Members

Protected Attributes inherited from [gdcm::PixmapReader](#)

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

12.155.1 Detailed Description

[ImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space.

See also

[Image](#)

Examples

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExplicitLittleEndian.cs](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

12.155.2 Constructor & Destructor Documentation

12.155.2.1 ImageReader()

gdcm::ImageReader::ImageReader ()

12.155.2.2 ~ImageReader()

gdcm::ImageReader::~~ImageReader () [override]

12.155.3 Member Function Documentation

12.155.3.1 GetImage() [1/2]

[Image](#) & gdcm::ImageReader::GetImage ()

12.155.3.2 GetImage() [2/2]

const [Image](#) & gdcm::ImageReader::GetImage () const

Return the read image.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExplicitLittleEndian.cs](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegionWithLUT.cs](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

12.155.3.3 Read()

```
bool gdcm::ImageReader::Read () [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an [Image](#).

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::ImageRegionReader](#).

Examples

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [DecompressImage.cs](#), [ExplicitLittleEndian.cs](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [ReadMultiTimesException.cxx](#), [RescaleImage.cs](#), and [threadgdcm.cxx](#).

12.155.3.4 ReadACRNEMAIImage()

```
bool gdcm::ImageReader::ReadACRNEMAIImage () [override], [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

12.155.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
    MediaStorage const & ms) [override], [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

The documentation for this class was generated from the following file:

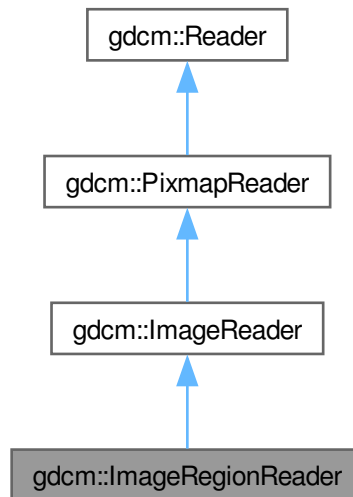
- [gdcmImageReader.h](#)

12.156 gdcm::ImageRegionReader Class Reference

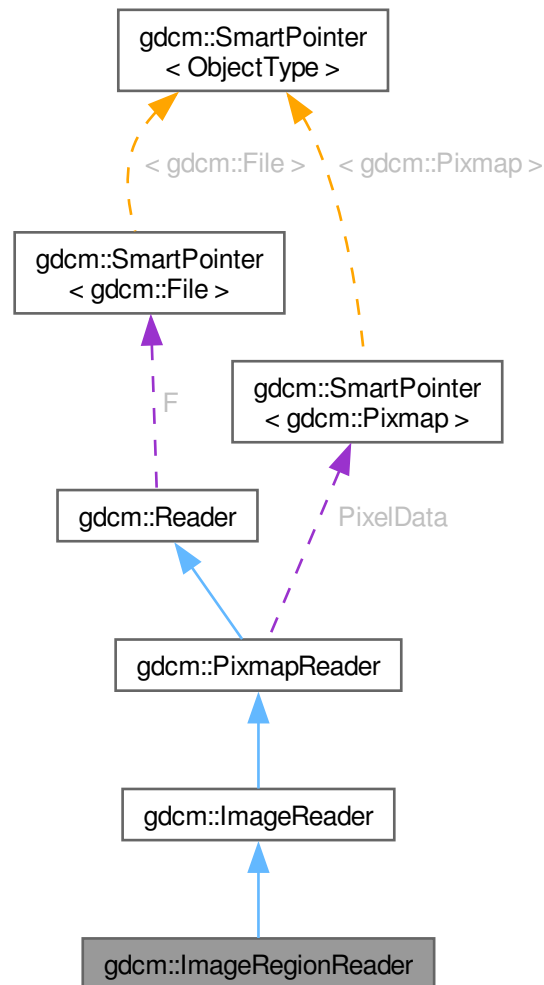
[ImageRegionReader](#).

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:



Collaboration diagram for `gdcm::ImageRegionReader`:



Public Member Functions

- `ImageRegionReader ()`
- `~ImageRegionReader ()` override
- `size_t ComputeBufferLength ()` const
- `Region` const & `GetRegion ()` const
- `bool ReadInformation ()`
- `bool ReadIntoBuffer (char *inreadbuffer, size_t buflen)`
- `void SetRegion (Region const ®ion)`
Set/Get `Region` to be read.

Public Member Functions inherited from [gdcm::ImageReader](#)

- [ImageReader](#) ()
- [~ImageReader](#) () override
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const

Return the read image.

Public Member Functions inherited from [gdcm::PixmapReader](#)

- [PixmapReader](#) ()
 - [~PixmapReader](#) () override
 - [Pixmap](#) & [GetPixmap](#) ()
 - const [Pixmap](#) & [GetPixmap](#) () const
- Return the read image (need to call [Read\(\)](#) first).
- bool [Read](#) () override

Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
 - virtual [~Reader](#) ()
 - bool [CanRead](#) () const
 - [File](#) & [GetFile](#) ()
- Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
- Set/Get [File](#).
- size_t [GetStreamCurrentPosition](#) () const
 - bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
- Will only read the specified selected private tags.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
- Will only read the specified selected tags.
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
 - void [SetFile](#) ([File](#) &file)
- Set/Get [File](#).
- void [SetFileName](#) (const char *filename__native)
 - void [SetStream](#) (std::istream &input__stream)
- Set the open-ed stream directly.

Protected Member Functions

- bool [Read](#) () override
- To prevent user from calling super class [Read\(\)](#) function.

Protected Member Functions inherited from [gdcm::ImageReader](#)

- bool [ReadACRNEMAIImage](#) () override
- bool [ReadImage](#) ([MediaStorage](#) const &ms) override

Protected Member Functions inherited from [gdcm::PixmapReader](#)

- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream * [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

Additional Inherited Members

Protected Attributes inherited from [gdcm::PixmapReader](#)

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

12.156.1 Detailed Description

[ImageRegionReader](#).

This class is able to read a region from a DICOM file containing an image. This implementation requires that the information stored in the DICOM header are consistent with what is in the encapsulated Pixel Data. This is technically not required by DICOM standard, which makes this implementation illegal with regards to the famous JPEG note: http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_8.2.html#para_4bcb841e-c6bf-4e26-82a5-3fad3c942da0

See also

[ImageReader](#)

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [TemplateEmptyImage.cxx](#).

12.156.2 Constructor & Destructor Documentation

12.156.2.1 [ImageRegionReader](#)()

[gdcm::ImageRegionReader::ImageRegionReader](#) ()

12.156.2.2 ~ImageRegionReader()

gdcm::ImageRegionReader::~~ImageRegionReader () [override]

12.156.3 Member Function Documentation

12.156.3.1 ComputeBufferLength()

size_t gdcm::ImageRegionReader::ComputeBufferLength () const

Explicit call which will compute the minimal buffer length that can hold the whole uncompressed image as defined by [Region](#) region.

Returns

0 upon error

12.156.3.2 GetRegion()

[Region](#) const & gdcm::ImageRegionReader::GetRegion () const

12.156.3.3 Read()

bool gdcm::ImageRegionReader::Read () [override], [protected], [virtual]

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

12.156.3.4 ReadInformation()

bool gdcm::ImageRegionReader::ReadInformation ()

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), and [TemplateEmptyImage.cxx](#).

12.156.3.5 ReadIntoBuffer()

```
bool gdcM::ImageRegionReader::ReadIntoBuffer (
    char * inreadbuffer,
    size_t buflen)
```

Read into buffer: For Python, the buflen param is deduced directly from the input bytearray passed as parameter (function only takes one param).

Returns

false upon error

Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

12.156.3.6 SetRegion()

```
void gdcM::ImageRegionReader::SetRegion (
    Region const & region)
```

Set/Get [Region](#) to be read.

Examples

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

The documentation for this class was generated from the following file:

- [gdcMImageRegionReader.h](#)

12.157 gdcM::ImageToImageFilter Class Reference

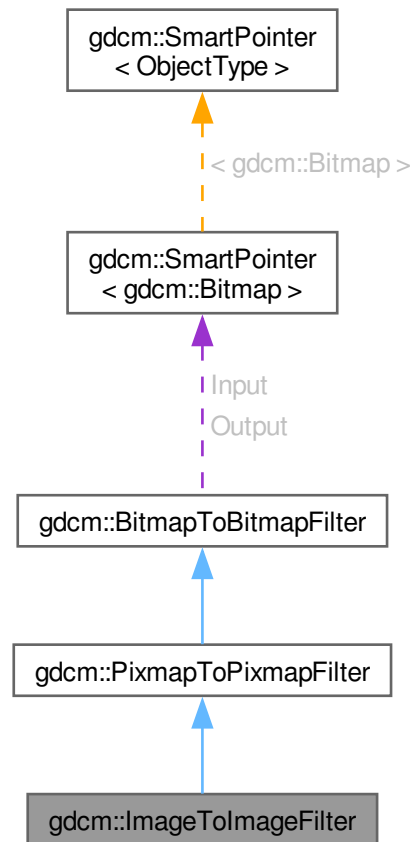
[ImageToImageFilter](#) class.

```
#include <gdcMImageToImageFilter.h>
```

Inheritance diagram for gdcM::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()=default
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const
Get Output image.

Public Member Functions inherited from [gdcm::PixmapToPixmapFilter](#)

- [PixmapToPixmapFilter](#) ()
- [~PixmapToPixmapFilter](#) ()=default
- [Pixmap](#) & [GetInput](#) ()
- const [Pixmap](#) & [GetOutput](#) () const
Get Output image.
- const [Pixmap](#) & [GetOutputAsPixmap](#) () const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter](#) ()
- [~BitmapToBitmapFilter](#) ()=default
- const [Bitmap](#) & [GetOutput](#) () const
Get Output image.
- const [Bitmap](#) & [GetOutputAsBitmap](#) () const
- void [SetInput](#) (const [Bitmap](#) &image)
Set input image.

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer](#)< [Bitmap](#) > [Input](#)
- [SmartPointer](#)< [Bitmap](#) > [Output](#)

12.157.1 Detailed Description

[ImageToImageFilter](#) class.

Super class for all filter taking an image and producing an output image

12.157.2 Constructor & Destructor Documentation

12.157.2.1 [ImageToImageFilter](#)()

[gdcm::ImageToImageFilter::ImageToImageFilter](#) ()

12.157.2.2 [~ImageToImageFilter](#)()

[gdcm::ImageToImageFilter::~~ImageToImageFilter](#) () [default]

12.157.3 Member Function Documentation

12.157.3.1 [GetInput](#)()

[Image](#) & [gdcm::ImageToImageFilter::GetInput](#) ()

12.157.3.2 GetOutput()

```
const Image & gdcm::ImageToImageFilter::GetOutput () const
```

Get Output image.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), and [ExplicitLittleEndian.cs](#).

The documentation for this class was generated from the following file:

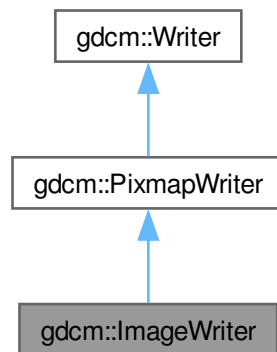
- [gdcmImageToImageFilter.h](#)

12.158 gdcm::ImageWriter Class Reference

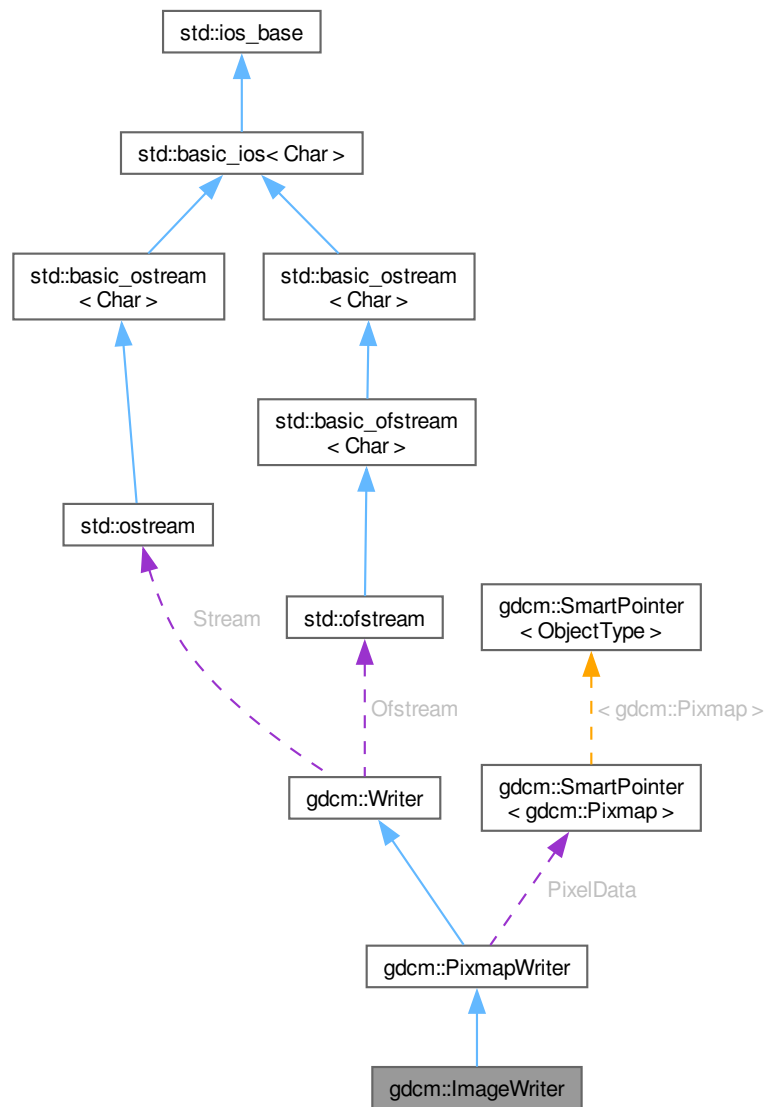
[ImageWriter](#).

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for `gdcm::ImageWriter`:



Public Member Functions

- `ImageWriter ()`
- `~ImageWriter ()` override
- `MediaStorage ComputeTargetMediaStorage ()`
- `const Image & GetImage ()` const override
- `Image & GetImage ()` override
- `bool Write ()` override

Write.

Public Member Functions inherited from [gdcm::PixmapWriter](#)

- [PixmapWriter](#) ()
 - [~PixmapWriter](#) () override
 - [Pixmap](#) & [GetPixmap](#) ()
 - const [Pixmap](#) & [GetPixmap](#) () const
 - virtual void [SetImage](#) ([Pixmap](#) const &img)
 - void [SetPixmap](#) ([Pixmap](#) const &img)
 - bool [Write](#) () override
- Write.

Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
 - virtual [~Writer](#) ()
 - void [CheckFileMetaInformationOff](#) ()
 - void [CheckFileMetaInformationOn](#) ()
 - [File](#) & [GetFile](#) ()
 - void [SetCheckFileMetaInformation](#) (bool b)
- Undocumented function, do not use (= leave default).
- void [SetFile](#) (const [File](#) &f)
- Set/Get the DICOM file ([DataSet](#) + Header).
- void [SetFileName](#) (const char *filename__native)
- Set the filename of DICOM file to write:
- void [SetStream](#) (std::ostream &output__stream)
- Set user ostream buffer.

Additional Inherited Members

Protected Member Functions inherited from [gdcm::PixmapWriter](#)

- void [DoIconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes inherited from [gdcm::PixmapWriter](#)

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

Protected Attributes inherited from [gdcm::Writer](#)

- `std::ofstream` * [Ofstream](#)
- `std::ostream` * [Stream](#)

12.158.1 Detailed Description

[ImageWriter](#).

This is an extended version of the [PixmapWriter](#). Pay attention that:

1. It will populate missing attribute for Secondary Capture [Image](#) Storage instances,
2. It may also change an input MR [Image](#) Storage instance into a pseudo Enhanced MR [Image](#) Storage instance whenever Modality LUT is required.
3. Some [DataElement](#) related to [gdcm::Image](#) may be slightly altered.

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [ExplicitLittleEndian.cs](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.158.2 Constructor & Destructor Documentation

12.158.2.1 [ImageWriter\(\)](#)

`gdcm::ImageWriter::ImageWriter ()`

12.158.2.2 [~ImageWriter\(\)](#)

`gdcm::ImageWriter::~~ImageWriter ()` [override]

12.158.3 Member Function Documentation

12.158.3.1 [ComputeTargetMediaStorage\(\)](#)

[MediaStorage](#) `gdcm::ImageWriter::ComputeTargetMediaStorage ()`

internal function used to compute a target [MediaStorage](#) the most appropriate User may want to call this function ahead of time (before Write)

Examples

[TemplateEmptyImage.cxx](#).

12.158.3.2 GetImage() [1/2]

const [Image](#) & gdcm::ImageWriter::GetImage () const [inline], [override], [virtual]

Set/Get [Image](#) to be written It will overwrite anything [Image](#) infos found in [DataSet](#) (see parent class to see how to pass dataset)

Reimplemented from [gdcm::PixmapWriter](#).

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.158.3.3 GetImage() [2/2]

[Image](#) & gdcm::ImageWriter::GetImage () [inline], [override], [virtual]

Reimplemented from [gdcm::PixmapWriter](#).

12.158.3.4 Write()

bool gdcm::ImageWriter::Write () [override], [virtual]

Write.

Reimplemented from [gdcm::Writer](#).

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmImageWriter.h](#)

12.159 gdcm::network::ImplementationClassUIDSub Class Reference

[ImplementationClassUIDSub](#).

```
#include <gdcmImplementationClassUIDSub.h>
```

Public Member Functions

- [ImplementationClassUIDSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.159.1 Detailed Description

[ImplementationClassUIDSub](#).

PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.159.2 Constructor & Destructor Documentation

12.159.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ()
```

12.159.3 Member Function Documentation

12.159.3.1 Print()

```
void gdcm::network::ImplementationClassUIDSub::Print (
    std::ostream & os) const
```

12.159.3.2 Read()

```
std::istream & gdcm::network::ImplementationClassUIDSub::Read (
    std::istream & is)
```

12.159.3.3 Size()

```
size_t gdcm::network::ImplementationClassUIDSub::Size () const
```

12.159.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationClassUIDSub::Write (
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmImplementationClassUIDSub.h](#)

12.160 gdcm::network::ImplementationUIDSub Class Reference

[ImplementationUIDSub.](#)

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

12.160.1 Detailed Description

[ImplementationUIDSub.](#)

[Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS \(A-ASSOCIATE-AC\)](#)

12.160.2 Constructor & Destructor Documentation

12.160.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ()
```

12.160.3 Member Function Documentation

12.160.3.1 Write()

```
const std::ostream & gdcm::network::ImplementationUIDSub::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmImplementationUIDSub.h](#)

12.161 gdcm::network::ImplementationVersionNameSub Class Reference

[ImplementationVersionNameSub.](#)

```
#include <gdcmImplementationVersionNameSub.h>
```

Public Member Functions

- [ImplementationVersionNameSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.161.1 Detailed Description

[ImplementationVersionNameSub](#).

[Table D.3-3](#) IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.161.2 Constructor & Destructor Documentation

12.161.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ()
```

12.161.3 Member Function Documentation

12.161.3.1 Print()

```
void gdcm::network::ImplementationVersionNameSub::Print (
    std::ostream & os) const
```

12.161.3.2 Read()

```
std::istream & gdcm::network::ImplementationVersionNameSub::Read (
    std::istream & is)
```

12.161.3.3 Size()

```
size_t gdcm::network::ImplementationVersionNameSub::Size () const
```

12.161.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationVersionNameSub::Write (
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

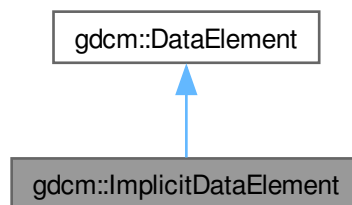
- [gdcmImplementationVersionNameSub.h](#)

12.162 gdcm::ImplicitDataElement Class Reference

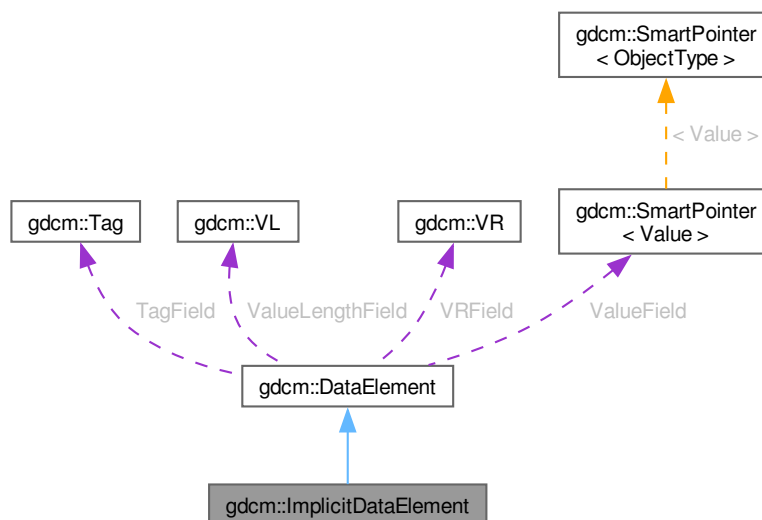
Class to represent an Implicit VR Data Element.

```
#include <gdcmImplicitDataElement.h>
```

Inheritance diagram for gdcm::ImplicitDataElement:



Collaboration diagram for gdcm::ImplicitDataElement:



Public Member Functions

- [VL GetLength \(\)](#) const

- `template<typename TSwap>`
`std::istream & Read (std::istream &is)`
- `template<typename TSwap>`
`std::istream & ReadPreValue (std::istream &is)`
- `template<typename TSwap>`
`std::istream & ReadValue (std::istream &is, bool readvalues=true)`
- `template<typename TSwap>`
`std::istream & ReadValueWithLength (std::istream &is, VL &length, bool readvalues=true)`
- `template<typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length, bool readvalues=true)`
- `template<typename TSwap>`
`const std::ostream & Write (std::ostream &os) const`

Public Member Functions inherited from `gdcm::DataElement`

- `DataElement` (const `DataElement` &_val)
- `DataElement` (const `Tag` &t=`Tag`(0), const `VL` &vl=0, const `VR` &vr=`VR::INVALID`)
- void `Clear` ()
Clear Data `Element` (make `Value` empty and invalidate `Tag` + `VR`).
- void `Empty` ()
Make Data `Element` empty (no `Value`).
- const `ByteValue` * `GetByteValue` () const
- `template<typename TDE>`
`VL GetLength` () const
- `SequenceOfFragments` * `GetSequenceOfFragments` ()
- const `SequenceOfFragments` * `GetSequenceOfFragments` () const
- `Tag` & `GetTag` ()
- const `Tag` & `GetTag` () const
Get `Tag`.
- `Value` & `GetValue` ()
- `Value` const & `GetValue` () const
Set/Get `Value` (bytes array, SQ of items, SQ of fragments):
- `SmartPointer< SequenceOfItems >` `GetValueAsSQ` () const
- `VL` & `GetVL` ()
- const `VL` & `GetVL` () const
Get `VL`.
- `VR` const & `GetVR` () const
- bool `IsEmpty` () const
Check if Data `Element` is empty.
- bool `IsUndefinedLength` () const
return if `Value` Length if of undefined length
- bool `operator<` (const `DataElement` &de) const
- `DataElement` & `operator=` (const `DataElement` &)=default
- bool `operator==` (const `DataElement` &de) const
- `template<typename TDE, typename TSwap>`
`std::istream & Read` (std::istream &is)
- `template<typename TDE, typename TSwap>`
`std::istream & ReadOrSkip` (std::istream &is, std::set< `Tag` > const &skiptags)

- `template<typename TDE, typename TSwap>`
`std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)`
- `template<typename TDE, typename TSwap>`
`std::istream & ReadWithLength (std::istream &is, VL &length)`
- `void SetByteValue (const char *array, VL length)`
- `void SetTag (const Tag &t)`
- `void SetValue (Value const &vl)`
- `void SetVL (const VL &vl)`
- `void SetVLToUndefined ()`
- `void SetVR (VR const &vr)`
- `template<typename TDE, typename TSwap>`
`const std::ostream & Write (std::ostream &os) const`

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- `typedef SmartPointer< Value > ValuePtr`

Protected Member Functions inherited from [gdcm::DataElement](#)

- `void SetValueFieldLength (VL vl, bool readvalues)`

Protected Attributes inherited from [gdcm::DataElement](#)

- `Tag TagField`
- `ValuePtr ValueField`
- `VL ValueLengthField`
- `VR VRField`

12.162.1 Detailed Description

Class to represent an Implicit [VR](#) Data [Element](#).

Note

bla

Examples

[ReadExplicitLengthSQIVR.cxx](#).

12.162.2 Member Function Documentation

12.162.2.1 GetLength()

[VL](#) `gdcm::ImplicitDataElement::GetLength () const`

12.162.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::Read (
    std::istream & is)
```

12.162.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.162.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```

12.162.2.5 ReadValueWithLength()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true)
```

12.162.2.6 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true)
```

12.162.2.7 Write()

```
template<typename TSwap>
const std::ostream & gdcm::ImplicitDataElement::Write (
    std::ostream & os) const
```

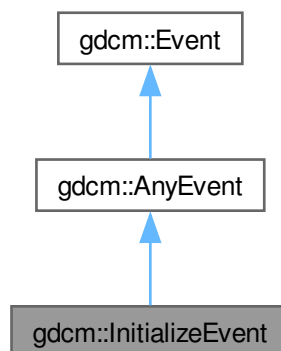
The documentation for this class was generated from the following file:

- [gdcmImplicitDataElement.h](#)

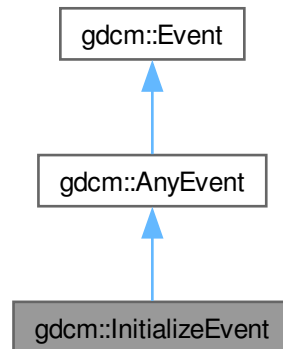
12.163 gdcm::InitializeEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for `gdcm::InitializeEvent`:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.164 gdcm::IOD Class Reference

Class for representing a [IOD](#).

```
#include <gdcmIOD.h>
```

Public Types

- typedef std::vector< [IODEntry](#) > [MapIODEntry](#)
- typedef MapIODEntry::size_type [SizeType](#)

Public Member Functions

- [IOD](#) ()=default
- void [AddIODEntry](#) (const [IODEntry](#) &iode)
- void [Clear](#) ()
- const [IODEntry](#) & [GetIODEntry](#) ([SizeType](#) idx) const
- [SizeType](#) [GetNumberOfIODs](#) () const
- [Type](#) [GetTypeFromTag](#) (const [Defs](#) &defs, const [Tag](#) &tag) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [IOD](#) &_val)

12.164.1 Detailed Description

Class for representing a [IOD](#).

Note

bla

See also

[Dict](#)

Examples

[TraverseModules.cxx](#).

12.164.2 Member Typedef Documentation

12.164.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

12.164.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

12.164.3 Constructor & Destructor Documentation

12.164.3.1 IOD()

```
gdcm::IOD::IOD () [default]
```

References [IOD\(\)](#), and [operator<<](#).

Referenced by [IOD\(\)](#), and [operator<<](#).

12.164.4 Member Function Documentation

12.164.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (  
    const IODEntry & iode) [inline]
```

12.164.4.2 Clear()

```
void gdcm::IOD::Clear () [inline]
```

12.164.4.3 GetIODEntry()

```
const IODEntry & gdcm::IOD::GetIODEntry (  
    SizeType idx) const [inline]
```

Examples

[TraverseModules.cxx](#).

12.164.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs () const [inline]
```

Examples

[TraverseModules.cxx](#).

12.164.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (  
    const Defs & defs,  
    const Tag & tag) const
```

12.164.5 Friends And Related Symbol Documentation

12.164.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const IOD & __val) [friend]
```

References [IOD\(\)](#).

Referenced by [IOD\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmIOD.h](#)

12.165 gdcmm::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmmIODEntry.h>
```

Public Member Functions

- [IODEntry](#) (const char *name="", const char *ref="", const char *inUsage="")
- const char * [GetIE](#) () const
- const char * [GetName](#) () const
- const char * [GetRef](#) () const
- const char * [GetUsage](#) () const
- [Usage::UsageType](#) [GetUsageType](#) () const
- void [SetIE](#) (const char *ie)
- void [SetName](#) (const char *name)
- void [SetRef](#) (const char *ref)
- void [SetUsage](#) (const char *inUsage)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [IODEntry](#) &_val)

12.165.1 Detailed Description

Class for representing a [IODEntry](#).

Note

A.1.3 [IOD Module Table](#) and Functional Group [Macro Table](#) This Section of each [IOD](#) defines in a tabular form the [Modules](#) comprising the [IOD](#). The following information must be specified for each [Module](#) in the table:

- The name of the [Module](#) or Functional Group
- A reference to the Section in Annex C which defines the [Module](#) or Functional Group
- The usage of the [Module](#) or Functional Group; whether it is:
 - Mandatory (see A.1.3.1) , abbreviated M
 - Conditional (see A.1.3.2) , abbreviated C
 - User Option (see A.1.3.3) , abbreviated U
- The [Modules](#) referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each [IOD](#), Mandatory [Modules](#) shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional [Modules](#) are Mandatory [Modules](#) if specific conditions are met. If the specified conditions are not met, this [Module](#) shall not be supported; that is, no information defined in that [Module](#) shall be sent. A.1.3.3 USER OPTION MODULES User Option [Modules](#) may or may not be supported. If an optional [Module](#) is supported, the [Attribute](#) Types specified in the [Modules](#) in Annex C shall be supported.

See also

[DictEntry](#)

Examples

[TraverseModules.cxx](#).

12.165.2 Constructor & Destructor Documentation

12.165.2.1 IODEntry()

```
gdcM::IODEntry::IODEntry (  
    const char * name = "",  
    const char * ref = "",  
    const char * inUsage = "") [inline]
```

Referenced by [operator<<](#).

12.165.3 Member Function Documentation

12.165.3.1 GetIE()

```
const char * gdcM::IODEntry::GetIE () const [inline]
```

12.165.3.2 GetName()

```
const char * gdcM::IODEntry::GetName () const [inline]
```

12.165.3.3 GetRef()

```
const char * gdcM::IODEntry::GetRef () const [inline]
```

Examples

[TraverseModules.cxx](#).

12.165.3.4 GetUsage()

```
const char * gdcM::IODEntry::GetUsage () const [inline]
```

12.165.3.5 GetUsageType()

```
Usage::UsageType gdcM::IODEntry::GetUsageType () const
```

12.165.3.6 SetIE()

```
void gdcM::IODEntry::SetIE (  
    const char * ie) [inline]
```

12.165.3.7 SetName()

```
void gdcm::IODEntry::SetName (
    const char * name) [inline]
```

12.165.3.8 SetRef()

```
void gdcm::IODEntry::SetRef (
    const char * ref) [inline]
```

12.165.3.9 SetUsage()

```
void gdcm::IODEntry::SetUsage (
    const char * inUsage) [inline]
```

12.165.4 Friends And Related Symbol Documentation

12.165.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const IODEntry & __val) [friend]
```

References [IODEntry\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmIODEntry.h](#)

12.166 gdcm::IODs Class Reference

Class for representing a [IODs](#).

```
#include <gdcmIODs.h>
```

Public Types

- typedef std::map< [IODName](#), [IOD](#) > [IODMapType](#)
- typedef [IODMapType](#)::const_iterator [IODMapTypeConstIterator](#)
- typedef std::string [IODName](#)

Public Member Functions

- [IODs](#) ()=default
- void [AddIOD](#) (const char *name, const [IOD](#) &module)
- [IODMapTypeConstIterator](#) [Begin](#) () const
- void [Clear](#) ()
- [IODMapTypeConstIterator](#) [End](#) () const
- const [IOD](#) & [GetIOD](#) (const char *name) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [IODs](#) &_val)

12.166.1 Detailed Description

Class for representing a [IODs](#).

Note

bla

See also

[IOD](#)

Examples

[TraverseModules.cxx](#).

12.166.2 Member Typedef Documentation

12.166.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

12.166.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const\_iterator gdcm::IODs::IODMapTypeConstIterator
```

Examples

[TraverseModules.cxx](#).

12.166.2.3 IODName

typedef std::string [gdcm::IODs::IODName](#)

Examples

[TraverseModules.cxx](#).

12.166.3 Constructor & Destructor Documentation

12.166.3.1 IODs()

gdcm::IODs::IODs () [default]

References [IODs\(\)](#), and [operator<<](#).

Referenced by [IODs\(\)](#), and [operator<<](#).

12.166.4 Member Function Documentation

12.166.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
    const char * name,
    const IOD & module) [inline]
```

12.166.4.2 Begin()

[IODMapTypeConstIterator](#) gdcm::IODs::Begin () const [inline]

Examples

[TraverseModules.cxx](#).

12.166.4.3 Clear()

void gdcm::IODs::Clear () [inline]

12.166.4.4 End()

[IODMapTypeConstIterator](#) gdcm::IODs::End () const [inline]

Examples

[TraverseModules.cxx](#).

12.166.4.5 GetIOD()

```
const IOD & gdcmm::IODs::GetIOD (  
    const char * name) const    [inline]
```

References [gdcmm_assert](#).

12.166.5 Friends And Related Symbol Documentation

12.166.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const IODs & __val)    [friend]
```

References [IODs\(\)](#).

Referenced by [IODs\(\)](#).

The documentation for this class was generated from the following file:

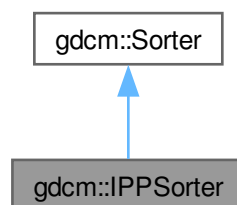
- [gdcmmIODs.h](#)

12.167 gdcmm::IPPSorter Class Reference

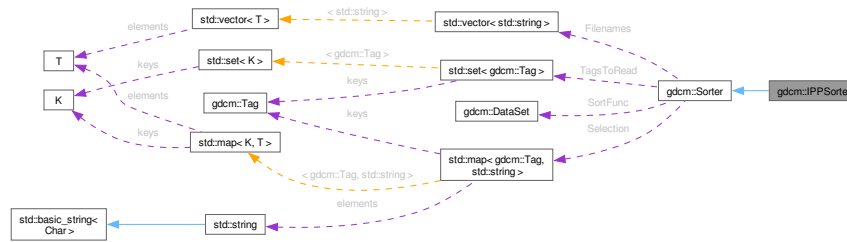
[IPPSorter](#).

```
#include <gdcmmIPPSorter.h>
```

Inheritance diagram for gdcmm::IPPSorter:



Collaboration diagram for gdcm::IPPSorter:



Public Member Functions

- [IPPSorter](#) ()
- double [GetDirectionCosinesTolerance](#) () const
- double [GetZSpacing](#) () const
- double [GetZSpacingTolerance](#) () const
- void [SetComputeZSpacing](#) (bool b)
- void [SetDirectionCosinesTolerance](#) (double tol)
- void [SetDropDuplicatePositions](#) (bool b)
- void [SetZSpacingTolerance](#) (double tol)
- bool [Sort](#) (std::vector< std::string > const &filenames) override

Public Member Functions inherited from [gdcm::Sorter](#)

- [Sorter](#) ()
- virtual [~Sorter](#) ()
- bool [AddSelect](#) ([Tag](#) const &tag, const char *value)
UNSUPPORTED FOR NOW.
- const std::vector< std::string > & [GetFileNames](#) () const
- void [Print](#) (std::ostream &os) const
Print.
- void [SetSortFunction](#) ([SortFunction](#) f)
- void [SetTagsToRead](#) (std::set< [Tag](#) > const &tags)
- virtual bool [StableSort](#) (std::vector< std::string > const &filenames)

Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

Protected Attributes inherited from [gdcm::Sorter](#)

- `std::vector< std::string >` [FileNames](#)
- `std::map< Tag, std::string >` [Selection](#)
- [SortFunction](#) [SortFunc](#)
- `std::set< Tag >` [TagsToRead](#)

Additional Inherited Members

Public Types inherited from [gdcm::Sorter](#)

- `typedef bool(* SortFunction) (DataSet const &, DataSet const &)`
Set the sort function which compares one dataset to the other.

Protected Types inherited from [gdcm::Sorter](#)

- `typedef std::map< Tag, std::string >` [SelectionMap](#)

12.167.1 Detailed Description

[IPPSorter](#).

Implement a simple [Image](#) Position ([Patient](#)) sorter, along the [Image Orientation](#) ([Patient](#)) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Warning

See special note for [SetZSpacingTolerance](#) when computing the ZSpacing from the IPP of each DICOM files (default tolerance for consistent spacing is: 1e-6mm)

For more information on [Spacing](#), and how it is defined in DICOM, advanced users may refers to:

http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing

[Bug](#) There are currently a couple of bugs in this implementation:

- Gantry Tilt is not considered (always an error)
- Application programmer should only sort valid [DataSet](#) (eg. [MRImageStorage](#), [CTImageStorage](#), [PETImageStorage](#))

Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), and [gdcmorthoplanes.cxx](#).

12.167.2 Constructor & Destructor Documentation

12.167.2.1 IPPSorter()

gdcm::IPPSorter::IPPSorter ()

12.167.3 Member Function Documentation

12.167.3.1 GetDirectionCosinesTolerance()

double gdcm::IPPSorter::GetDirectionCosinesTolerance () const [inline]

References [DirCosTolerance](#).

12.167.3.2 GetZSpacing()

double gdcm::IPPSorter::GetZSpacing () const [inline]

Read-only function to provide access to the computed value for the Z-Spacing The ComputeZSpacing must have been set to true before execution of sort algorithm. Call this function after calling [Sort\(\)](#); Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

Examples

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

References [ZSpacing](#).

12.167.3.3 GetZSpacingTolerance()

double gdcm::IPPSorter::GetZSpacingTolerance () const [inline]

References [ZTolerance](#).

12.167.3.4 SetComputeZSpacing()

```
void gdcmm::IPPSorter::SetComputeZSpacing (
    bool b) [inline]
```

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the [Image Position \(Patient\)](#) Potential reason for failure:

1. ALL slices are taken into account, if one slice is missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the [Series](#)

Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

References [ComputeZSpacing](#).

12.167.3.5 SetDirectionCosinesTolerance()

```
void gdcmm::IPPSorter::SetDirectionCosinesTolerance (
    double tol) [inline]
```

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426953\\0.00369275\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.00369272\\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the distance in between 1.0 to the dot product of those cross vectors. In a perfect world this dot product is of course 1.0 which imply a [DirectionCosines](#) tolerance of exactly 0.0 (default).

References [DirCosTolerance](#).

12.167.3.6 SetDropDuplicatePositions()

```
void gdcmm::IPPSorter::SetDropDuplicatePositions (
    bool b) [inline]
```

Makes the [IPPSorter](#) ignore multiple images located at the same position. Only the first occurrence will be kept. DropDuplicatePositions defaults to false.

References [DropDuplicatePositions](#).

12.167.3.7 SetZSpacingTolerance()

```
void gdcmm::IPPSorter::SetZSpacingTolerance (
    double tol) [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the series, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples

[Compute3DSpacing.cxx](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

References [ZTolerance](#).

12.167.3.8 Sort()

```
bool gdcm::IPPSorter::Sort (  
    std::vector< std::string > const & filenames)  [override], [virtual]
```

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacingTolerance, ...) Return value indicate if sorting could be achieved,. Warning this does NOT imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from [gdcm::Sorter](#).

Examples

[Compute3DSpacing.cxx](#), [VolumeSorter.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

12.167.4 Member Data Documentation

12.167.4.1 ComputeZSpacing

```
bool gdcm::IPPSorter::ComputeZSpacing  [protected]
```

Referenced by [SetComputeZSpacing\(\)](#).

12.167.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance  [protected]
```

Referenced by [GetDirectionCosinesTolerance\(\)](#), and [SetDirectionCosinesTolerance\(\)](#).

12.167.4.3 DropDuplicatePositions

```
bool gdcm::IPPSorter::DropDuplicatePositions  [protected]
```

Referenced by [SetDropDuplicatePositions\(\)](#).

12.167.4.4 ZSpacing

```
double gdcm::IPPSorter::ZSpacing  [protected]
```

Referenced by [GetZSpacing\(\)](#).

12.167.4.5 ZTolerance

double gdcM::IPPSorter::ZTolerance [protected]

Referenced by [GetZSpacingTolerance\(\)](#), and [SetZSpacingTolerance\(\)](#).

The documentation for this class was generated from the following file:

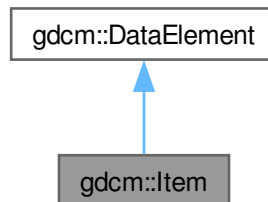
- [gdcMIPPSorter.h](#)

12.168 gdcM::Item Class Reference

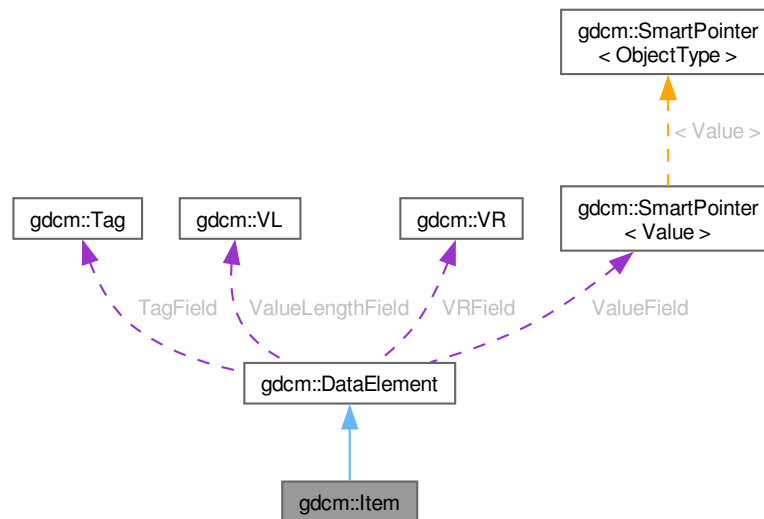
Class to represent an [Item](#).

```
#include <gdcMItem.h>
```

Inheritance diagram for gdcM::Item:



Collaboration diagram for gdcm::Item:



Public Member Functions

- [Item](#) ()
- [Item](#) (Item const &val)
- void [Clear](#) ()
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [Tag](#) &t) const
- template<typename TDE>
 [VL](#) [GetLength](#) () const
- [DataSet](#) & [GetNestedDataSet](#) ()
- const [DataSet](#) & [GetNestedDataSet](#) () const
- void [InsertDataElement](#) (const [DataElement](#) &de)
- template<typename TDE, typename TSwap>
 std::istream & [Read](#) (std::istream &is)
- void [SetNestedDataSet](#) (const [DataSet](#) &nested)
- template<typename TDE, typename TSwap>
 const std::ostream & [Write](#) (std::ostream &os) const

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
 Clear [Data Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()

- Make Data [Element](#) empty (no [Value](#)).
- [const ByteValue * GetByteValue \(\) const](#)
- [template<typename TDE> VL GetLength \(\) const](#)
- [SequenceOfFragments * GetSequenceOfFragments \(\)](#)
- [const SequenceOfFragments * GetSequenceOfFragments \(\) const](#)
- [Tag & GetTag \(\)](#)
- [const Tag & GetTag \(\) const](#)
- Get [Tag](#).
- [Value & GetValue \(\)](#)
- [Value const & GetValue \(\) const](#)
- Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer< SequenceOfItems > GetValueAsSQ \(\) const](#)
- [VL & GetVL \(\)](#)
- [const VL & GetVL \(\) const](#)
- Get [VL](#).
- [VR const & GetVR \(\) const](#)
- [bool IsEmpty \(\) const](#)
- Check if Data [Element](#) is empty.
- [bool IsUndefinedLength \(\) const](#)
- return if [Value](#) Length if of undefined length
- [bool operator< \(const DataElement &de\) const](#)
- [DataElement & operator= \(const DataElement &\)=default](#)
- [bool operator== \(const DataElement &de\) const](#)
- [template<typename TDE, typename TSwap> std::istream & Read \(std::istream &is\)](#)
- [template<typename TDE, typename TSwap> std::istream & ReadOrSkip \(std::istream &is, std::set< Tag > const &skiptags\)](#)
- [template<typename TDE, typename TSwap> std::istream & ReadPreValue \(std::istream &is, std::set< Tag > const &skiptags\)](#)
- [template<typename TDE, typename TSwap> std::istream & ReadValue \(std::istream &is, std::set< Tag > const &skiptags\)](#)
- [template<typename TDE, typename TSwap> std::istream & ReadValueWithLength \(std::istream &is, VL &length, std::set< Tag > const &skiptags\)](#)
- [template<typename TDE, typename TSwap> std::istream & ReadWithLength \(std::istream &is, VL &length\)](#)
- [void SetByteValue \(const char *array, VL length\)](#)
- [void SetTag \(const Tag &t\)](#)
- [void SetValue \(Value const &vl\)](#)
- [void SetVL \(const VL &vl\)](#)
- [void SetVLToUndefined \(\)](#)
- [void SetVR \(VR const &vr\)](#)
- [template<typename TDE, typename TSwap> const std::ostream & Write \(std::ostream &os\) const](#)

Friends

- [std::ostream & operator<< \(std::ostream &os, const Item &val\)](#)

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.168.1 Detailed Description

Class to represent an [Item](#).

A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of [VR](#) SQ shall be encoded as a DICOM Standard Data [Element](#) with a specific Data [Element](#) [Tag](#) of [Value](#) (FFFE,E000). The [Item](#) [Tag](#) is followed by a 4 byte [Item](#) Length field encoded in one of the following two ways Explicit/Implicit

Note

ITEM: A component of the [Value](#) of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set.

Examples

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [SimplePrint.cs](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

12.168.2 Constructor & Destructor Documentation

12.168.2.1 [Item](#)() [1/2]

[gdcm::Item::Item](#) () [inline]

References [gdcm::DataElement::DataElement\(\)](#).

Referenced by [Item\(\)](#), and [operator<<](#).

12.168.2.2 Item() [2/2]

```
gdcmm::Item::Item (  
    Item const & val) [inline]
```

References [gdcmm::DataElement::DataElement\(\)](#), and [Item\(\)](#).

12.168.3 Member Function Documentation

12.168.3.1 Clear()

```
void gdcmm::Item::Clear () [inline]
```

References [gdcmm::DataElement::Clear\(\)](#).

Referenced by [gdcmm::SequenceOfItems::Read\(\)](#).

12.168.3.2 FindDataElement()

```
bool gdcmm::Item::FindDataElement (  
    const Tag & t) const [inline]
```

12.168.3.3 GetDataElement()

```
const DataElement & gdcmm::Item::GetDataElement (  
    const Tag & t) const [inline]
```

References [gdcmm::DataElement::DataElement\(\)](#).

12.168.3.4 GetLength()

```
template<typename TDE>  
VL gdcmm::Item::GetLength () const
```

12.168.3.5 GetNestedDataSet() [1/2]

```
DataSet & gdcmm::Item::GetNestedDataSet () [inline]
```

12.168.3.6 GetNestedDataSet() [2/2]

```
const DataSet & gdcm::Item::GetNestedDataSet () const [inline]
```

Examples

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [SimplePrint.cs](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

Referenced by [gdcm::SequenceOfItems::Read\(\)](#).

12.168.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
    const DataElement & de) [inline]
```

References [gdcm::DataElement::DataElement\(\)](#), [gdcm_assert](#), and [gdcm::DataElement::IsUndefinedLength\(\)](#).

12.168.3.8 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::Item::Read (
    std::istream & is) [inline]
```

References [gdcm::ByteSwapFilter::ByteSwap\(\)](#), [gdcm::DataSet::Clear\(\)](#), [gdcm_assert](#), [gdcmDebugMacro](#), [gdcmErrorMacro](#), [gdcmWarningMacro](#), [gdcm::DataSet::IsEmpty\(\)](#), [gdcm::DataElement::ReadWithLength\(\)](#), [gdcm::ByteSwapFilter::SetByteSwapTag\(\)](#), [gdcm::SwapperDoOp::Swap\(\)](#), [gdcm::DataElement::TagField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [gdcm::SequenceOfItems::Read\(\)](#).

12.168.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
    const DataSet & nested) [inline]
```

12.168.3.10 Write()

```
template<typename TDE, typename TSwap>
const std::ostream & gdcm::Item::Write (
    std::ostream & os) const [inline]
```

References [gdcm_assert](#), [gdcmWarningMacro](#), [gdcm::VL::GetLength\(\)](#), [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueLengthField](#), [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

12.168.4 Friends And Related Symbol Documentation

12.168.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Item & val) [friend]
```

References [Item\(\)](#), [operator<<](#), [gdcmm::DataSet::Print\(\)](#), [gdcmm::DataElement::TagField](#), and [gdcmm::DataElement::ValueLength](#).

Referenced by [operator<<](#).

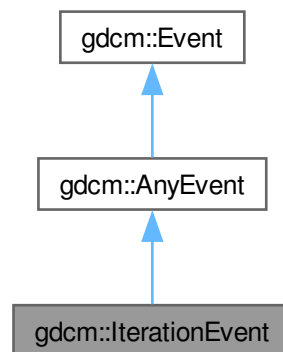
The documentation for this class was generated from the following file:

- [gdcmmItem.h](#)

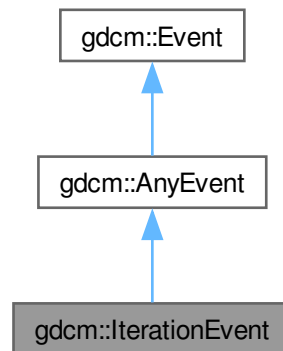
12.169 gdcmm::IterationEvent Class Reference

```
#include <gdcmmEvent.h>
```

Inheritance diagram for gdcmm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

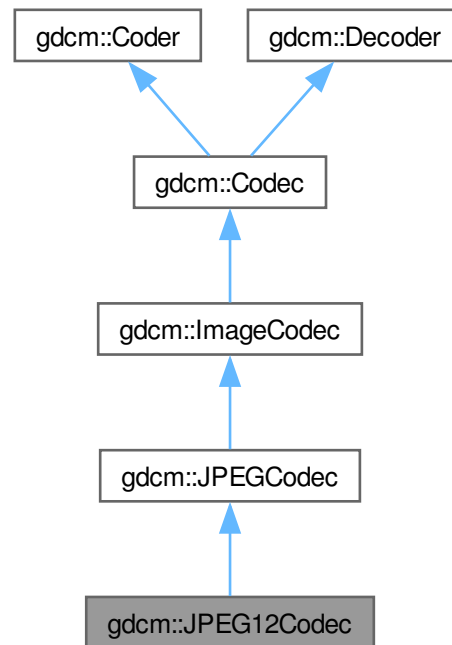
- [gdcmEvent.h](#)

12.170 gdcm::JPEG12Codec Class Reference

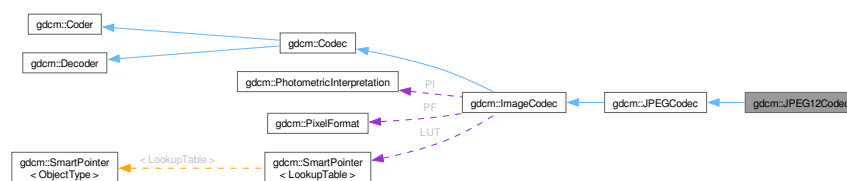
Class to do JPEG 12bits (lossy & lossless).

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for `gdcm::JPEG12Codec`:



Collaboration diagram for `gdcm::JPEG12Codec`:



Public Member Functions

- `JPEG12Codec ()`
- `~JPEG12Codec ()` override
- `bool DecodeByStreams (std::istream &is, std::ostream &os)` override
- `bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)` override
- `bool InternalCode (const char *input, unsigned long len, std::ostream &os)` override

Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Compress into JPEG.
- void [ComputeOffsetTable](#) (bool b)
Compute the offset table:
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen) override
- bool [IsStateSuspension](#) () const override

Protected Member Functions inherited from [gdcm::JPEGCodec](#)

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.170.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless).

Note

internal class

12.170.2 Constructor & Destructor Documentation

12.170.2.1 JPEG12Codec()

`gdcm::JPEG12Codec::JPEG12Codec ()`

12.170.2.2 ~JPEG12Codec()

`gdcm::JPEG12Codec::~~JPEG12Codec ()` [override]

12.170.3 Member Function Documentation

12.170.3.1 DecodeByStreams()

`bool gdcm::JPEG12Codec::DecodeByStreams (`
 `std::istream & is,`
 `std::ostream & os) [override], [virtual]`

Reimplemented from [gdcm::ImageCodec](#).

12.170.3.2 EncodeBuffer()

```
bool gdcm::JPEG12Codec::EncodeBuffer (  
    std::ostream & os,  
    const char * data,  
    size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

12.170.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (  
    std::istream & is,  
    TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.170.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (  
    const char * input,  
    unsigned long len,  
    std::ostream & os)  [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

12.170.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension () const  [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

The documentation for this class was generated from the following file:

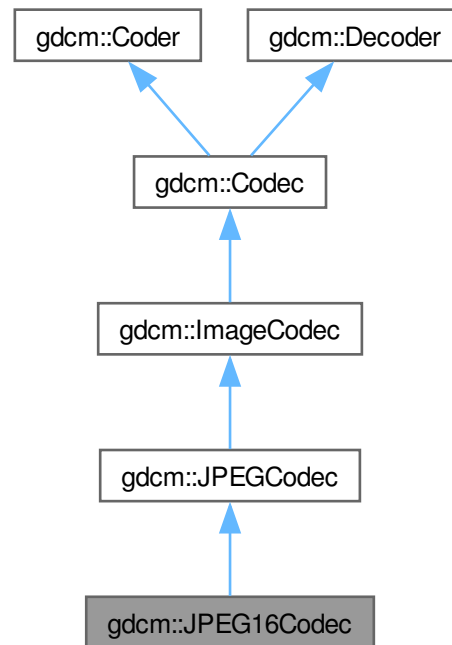
- [gdcmJPEG12Codec.h](#)

12.171 gdcm::JPEG16Codec Class Reference

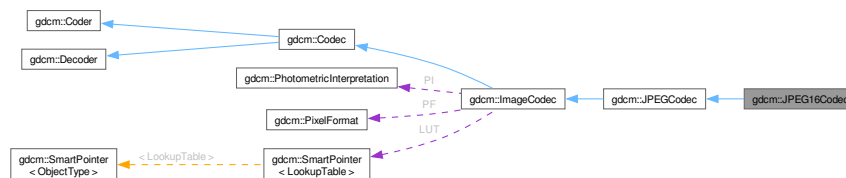
Class to do JPEG 16bits (lossless).

```
#include <gdcmJPEG16Codec.h>
```

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- [JPEG16Codec](#) ()
- [~JPEG16Codec](#) () override
- [bool DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- [bool GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- [bool InternalCode](#) (const char *input, unsigned long len, std::ostream &os) override

Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Compress into JPEG.
- void [ComputeOffsetTable](#) (bool b)
Compute the offset table:
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen) override
- bool [IsStateSuspension](#) () const override

Protected Member Functions inherited from [gdcm::JPEGCodec](#)

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.171.1 Detailed Description

Class to do JPEG 16bits (lossless).

Note

internal class

12.171.2 Constructor & Destructor Documentation

12.171.2.1 JPEG16Codec()

`gdcm::JPEG16Codec::JPEG16Codec ()`

12.171.2.2 ~JPEG16Codec()

`gdcm::JPEG16Codec::~~JPEG16Codec ()` [override]

12.171.3 Member Function Documentation

12.171.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.171.3.2 EncodeBuffer()

```
bool gdcm::JPEG16Codec::EncodeBuffer (  
    std::ostream & os,  
    const char * data,  
    size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

12.171.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (  
    std::istream & is,  
    TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.171.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (  
    const char * input,  
    unsigned long len,  
    std::ostream & os)  [override], [virtual]
```

Reimplemented from [gdcm::Coder](#).

12.171.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension () const  [override], [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

The documentation for this class was generated from the following file:

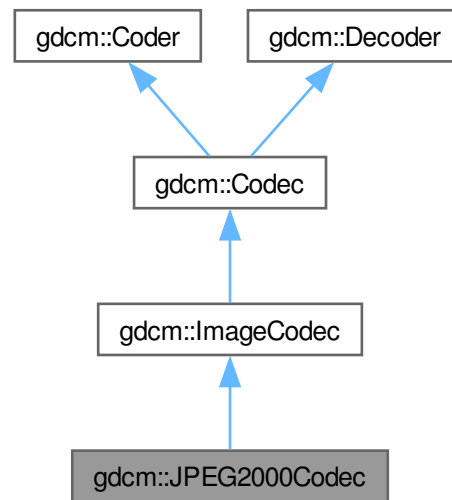
- [gdcmJPEG16Codec.h](#)

12.172 gdcM::JPEG2000Codec Class Reference

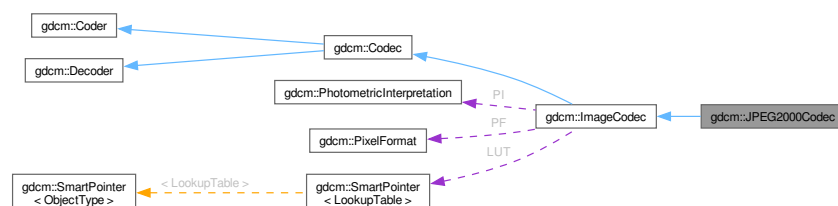
Class to do JPEG 2000.

```
#include <gdcMJPEG2000Codec.h>
```

Inheritance diagram for gdcM::JPEG2000Codec:



Collaboration diagram for gdcM::JPEG2000Codec:



Public Member Functions

- [JPEG2000Codec](#) ()
- [~JPEG2000Codec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- double [GetQuality](#) (unsigned int idx=0) const
- double [GetRate](#) (unsigned int idx=0) const
- void [SetMCT](#) (unsigned int mct)
- void [SetNumberOfResolutions](#) (unsigned int nres)
- void [SetNumberOfThreadsForDecompression](#) (int nThreads)
- void [SetQuality](#) (unsigned int idx, double q)
- void [SetRate](#) (unsigned int idx, double rate)
- void [SetReversible](#) (bool res)
- void [SetTileSize](#) (unsigned int tx, unsigned int ty)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.172.1 Detailed Description

Class to do JPEG 2000.

Note

the class will produce JPC (JPEG 2000 codestream), since some private implementor are using full jp2 file the decoder tolerate jp2 input this is an implementation of an [ImageCodec](#)

12.172.2 Constructor & Destructor Documentation

12.172.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ()
```

12.172.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~~JPEG2000Codec () [override]
```

12.172.3 Member Function Documentation

12.172.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
    TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
    TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.5 Clone()

```
ImageCodec * gdcm::JPEG2000Codec::Clone () const  [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.172.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
    DataElement const & in_,
    DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

12.172.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.9 DecodeExtent()

```
bool gdcm::JPEG2000Codec::DecodeExtent (
    char * buffer,
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax,
    std::istream & is) [protected]
```

12.172.3.10 GetHeaderInfo()

```
bool gdcm::JPEG2000Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

Referenced by [StopEncode\(\)](#).

12.172.3.11 GetQuality()

```
double gdcm::JPEG2000Codec::GetQuality (
    unsigned int idx = 0) const
```

12.172.3.12 GetRate()

```
double gdcM::JPEG2000Codec::GetRate (
    unsigned int idx = 0) const
```

12.172.3.13 IsFrameEncoder()

```
bool gdcM::JPEG2000Codec::IsFrameEncoder () [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

12.172.3.14 IsRowEncoder()

```
bool gdcM::JPEG2000Codec::IsRowEncoder () [override], [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

12.172.3.15 SetMCT()

```
void gdcM::JPEG2000Codec::SetMCT (
    unsigned int mct)
```

12.172.3.16 SetNumberOfResolutions()

```
void gdcM::JPEG2000Codec::SetNumberOfResolutions (
    unsigned int nres)
```

12.172.3.17 SetNumberOfThreadsForDecompression()

```
void gdcM::JPEG2000Codec::SetNumberOfThreadsForDecompression (
    int nThreads)
```

Set Number of threads

Parameters

nThreads	: number of threads for decompression codec, if 0 or 1 decompression is done in current thread, if negative value is set determine how many virtual threads are available
----------	---

12.172.3.18 SetQuality()

```
void gdcM::JPEG2000Codec::SetQuality (
    unsigned int idx,
    double q)
```

12.172.3.19 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
    unsigned int idx,
    double rate)
```

12.172.3.20 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
    bool res)
```

12.172.3.21 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
    unsigned int tx,
    unsigned int ty)
```

12.172.3.22 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.172.3.23 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

References [GetHeaderInfo\(\)](#).

12.172.4 Friends And Related Symbol Documentation

12.172.4.1 Bitmap

```
friend class Bitmap [friend]
```

References [Bitmap](#).

Referenced by [Bitmap](#).

12.172.4.2 ImageRegionReader

friend class ImageRegionReader [friend]

References [ImageRegionReader](#).

Referenced by [ImageRegionReader](#).

The documentation for this class was generated from the following file:

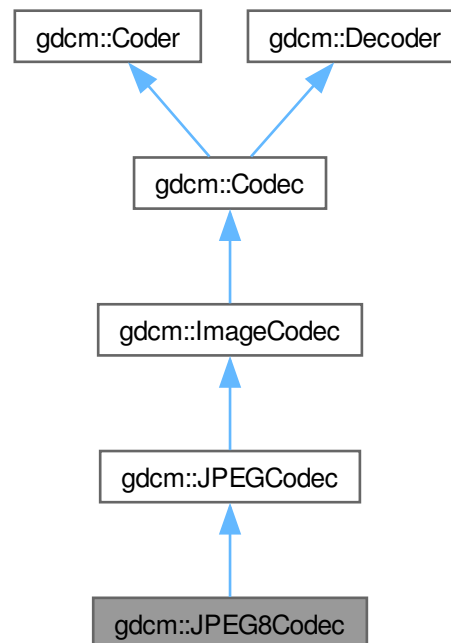
- [gdcmJPEG2000Codec.h](#)

12.173 gdcm::JPEG8Codec Class Reference

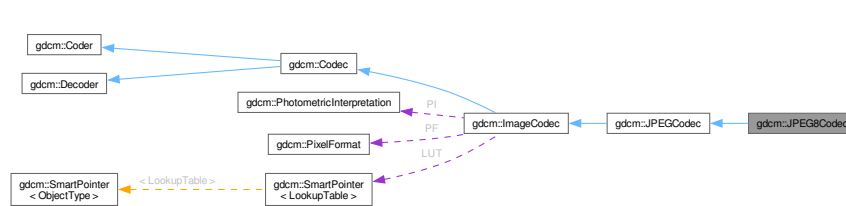
Class to do JPEG 8bits (lossy & lossless).

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- [JPEG8Codec](#) ()
- [~JPEG8Codec](#) () override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [InternalCode](#) (const char *input, unsigned long len, std::ostream &os) override

Public Member Functions inherited from [gdcm::JPEGCodec](#)

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Compress into JPEG.
- void [ComputeOffsetTable](#) (bool b)
Compute the offset table:
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [EncodeBuffer](#) (std::ostream &os, const char *data, size_t datalen) override
- bool [IsStateSuspension](#) () const override

Protected Member Functions inherited from [gdcm::JPEGCodec](#)

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

Protected Attributes inherited from [gdcm::JPEGCodec](#)

- int [BitSample](#)
- int [Quality](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.173.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless).

Note

internal class

12.173.2 Constructor & Destructor Documentation

12.173.2.1 JPEG8Codec()

`gdcm::JPEG8Codec::JPEG8Codec ()`

12.173.2.2 ~JPEG8Codec()

`gdcm::JPEG8Codec::~~JPEG8Codec ()` [override]

12.173.3 Member Function Documentation

12.173.3.1 DecodeByStreams()

`bool gdcm::JPEG8Codec::DecodeByStreams (`
 `std::istream & is,`
 `std::ostream & os)` [override], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.173.3.2 EncodeBuffer()

`bool gdcm::JPEG8Codec::EncodeBuffer (`
 `std::ostream & os,`
 `const char * data,`
 `size_t datalen)` [override], [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

12.173.3.3 GetHeaderInfo()

`bool gdcm::JPEG8Codec::GetHeaderInfo (`
 `std::istream & is,`
 [TransferSyntax](#) & ts) [override], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.173.3.4 InternalCode()

`bool gdcm::JPEG8Codec::InternalCode (`
 `const char * input,`
 `unsigned long len,`
 `std::ostream & os)` [override], [virtual]

Reimplemented from [gdcm::Coder](#).

12.173.3.5 IsStateSuspension()

bool gdcm::JPEG8Codec::IsStateSuspension () const [override], [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

The documentation for this class was generated from the following file:

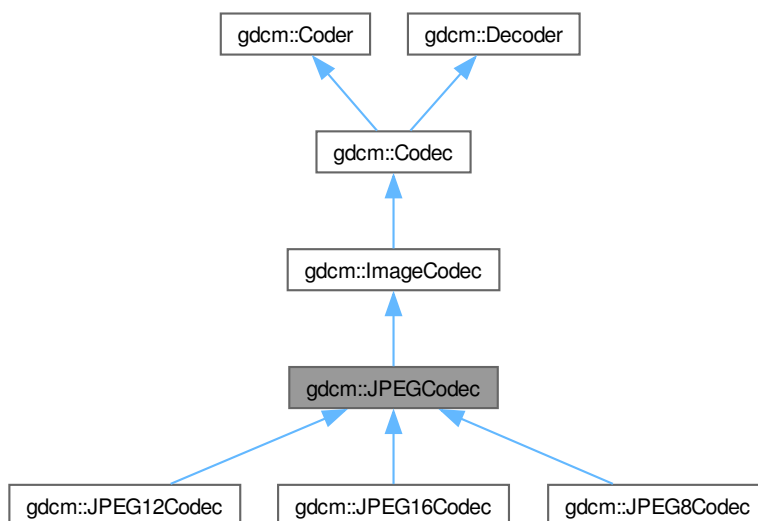
- [gdcmJPEG8Codec.h](#)

12.174 gdcm::JPEGCodec Class Reference

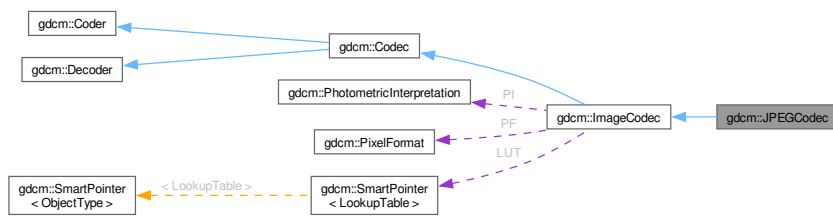
JPEG codec.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- [JPEGCodec](#) ()
- [~JPEGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Compress into JPEG.
- void [ComputeOffsetTable](#) (bool b)
Compute the offset table:
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- virtual bool [EncodeBuffer](#) (std::ostream &out, const char *inbuffer, size_t inlen)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf) override
- void [SetQuality](#) (double q)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi) override
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes

- int [BitSample](#)
- int [Quality](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

12.174.1 Detailed Description

JPEG codec.

Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [JPEG8Codec](#), [JPEG12Codec](#) & [JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f2080e1
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a6243
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f093ed
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c921f

Examples

[CompressLossyJPEG.cs](#), [FileChangeTSLossy.cs](#), and [GetJPEGSamplePrecision.cxx](#).

12.174.2 Constructor & Destructor Documentation

12.174.2.1 JPEGCodec()

gdcm::JPEGCodec::JPEGCodec ()

12.174.2.2 ~JPEGCodec()

gdcm::JPEGCodec::~~JPEGCodec () [override]

12.174.3 Member Function Documentation

12.174.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

Examples

[CompressLossyJPEG.cs](#).

12.174.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
    TransferSyntax const & ) const    [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.5 Clone()

```
ImageCodec * gdcm::JPEGCodec::Clone () const    [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.174.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
    DataElement const & in,
    DataElement & out)    [override], [virtual]
```

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

12.174.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
    bool b)
```

Compute the offset table:

12.174.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
    DataElement const & ,
    DataElement & )    [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.10 DecodeExtent()

```
bool gdcm::JPEGCodec::DecodeExtent (
    char * buffer,
    unsigned int xmin,
    unsigned int xmax,
    unsigned int ymin,
    unsigned int ymax,
    unsigned int zmin,
    unsigned int zmax,
    std::istream & is)  [protected]
```

12.174.3.11 EncodeBuffer()

```
virtual bool gdcm::JPEGCodec::EncodeBuffer (
    std::ostream & out,
    const char * inbuffer,
    size_t inlen)  [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

12.174.3.12 GetHeaderInfo()

```
bool gdcm::JPEGCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

Examples

[GetJPEGSamplePrecision.cxx](#).

12.174.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless () const
```

12.174.3.14 GetQuality()

double gdcm::JPEGCodec::GetQuality () const

12.174.3.15 IsFrameEncoder()

bool gdcm::JPEGCodec::IsFrameEncoder () [override], [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.16 IsRowEncoder()

bool gdcm::JPEGCodec::IsRowEncoder () [override], [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.17 IsStateSuspension()

virtual bool gdcm::JPEGCodec::IsStateSuspension () const [protected], [virtual]

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

12.174.3.18 IsValid()

bool gdcm::JPEGCodec::IsValid (
 [PhotometricInterpretation](#) const & pi) [override], [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.19 SetBitSample()

void gdcm::JPEGCodec::SetBitSample (
 int bit) [protected]

12.174.3.20 SetLossless()

void gdcm::JPEGCodec::SetLossless (
 bool l)

Examples

[CompressLossyJPEG.cs](#), and [FileChangeTSLossy.cs](#).

12.174.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
    PixelFormat const & pf) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

Examples

[GetJPEGSamplePrecision.cxx](#).

12.174.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
    double q)
```

Examples

[CompressLossyJPEG.cs](#), and [FileChangeTSLossy.cs](#).

12.174.3.23 StartEncode()

```
bool gdcm::JPEGCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.174.3.24 StopEncode()

```
bool gdcm::JPEGCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.174.4 Friends And Related Symbol Documentation

12.174.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

References [ImageRegionReader](#).

Referenced by [ImageRegionReader](#).

12.174.5 Member Data Documentation

12.174.5.1 BitSample

int gdcM::JPEGCodec::BitSample [protected]

12.174.5.2 Quality

int gdcM::JPEGCodec::Quality [protected]

The documentation for this class was generated from the following file:

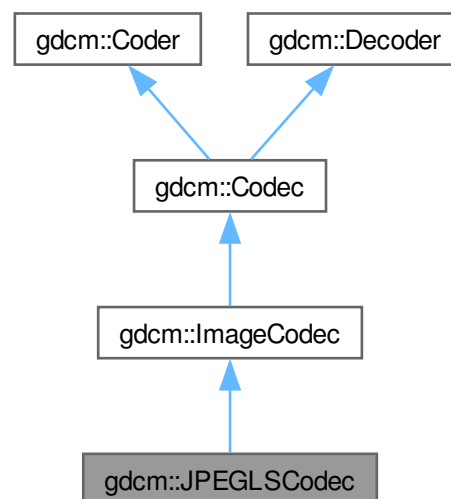
- [gdcMJPEGCodec.h](#)

12.175 gdcM::JPEGLSCodec Class Reference

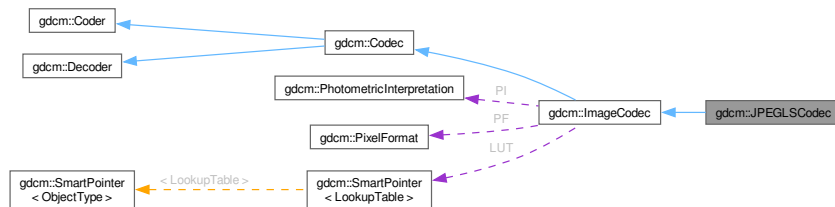
JPEG-LS.

```
#include <gdcMJPEGLSCodec.h>
```

Inheritance diagram for gdcM::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- [JPEGLSCodec](#) ()
- [~JPEGLSCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Code.
- bool [Decode](#) ([DataElement](#) const &in, char *outBuffer, size_t inBufferLength, uint32_t inXMin, uint32_t inXMax, uint32_t inYMin, uint32_t inYMax, uint32_t inZMin, uint32_t inZMax)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [GetLossless](#) () const
- void [SetBufferLength](#) (unsigned long l)
- void [SetLossless](#) (bool l)
- void [SetLossyError](#) (int error)
[0-3] generally

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const

- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.175.1 Detailed Description

JPEG-LS.

Note

codec that implement the JPEG-LS compression this is an implementation of [ImageCodec](#) for JPEG-LS

It uses the CharLS JPEG-LS implementation <https://github.com/team-charls/charls>

12.175.2 Constructor & Destructor Documentation

12.175.2.1 JPEGLSCodec()

gdcm::JPEGLSCodec::JPEGLSCodec ()

12.175.2.2 ~JPEGLSCodec()

`gdcm::JPEGLSCodec::~~JPEGLSCodec ()` [override]

12.175.3 Member Function Documentation

12.175.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (  
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (  
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.5 Clone()

[ImageCodec](#) * gdcm::JPEGLSCodec::Clone () const [override], [virtual]

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.175.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (  
    DataElement const & in_,  
    DataElement & out_) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

12.175.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (  
    DataElement const & in,  
    char * outBuffer,  
    size_t inBufferLength,  
    uint32_t inXMin,  
    uint32_t inXMax,  
    uint32_t inYMin,  
    uint32_t inYMax,  
    uint32_t inZMin,  
    uint32_t inZMax)
```

12.175.3.8 Decode() [2/2]

```
bool gdcm::JPEGLSCodec::Decode (  
    DataElement const & ,  
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.9 DecodeExtent()

```
bool gdcm::JPEGLSCodec::DecodeExtent (  
    char * buffer,  
    unsigned int xmin,  
    unsigned int xmax,  
    unsigned int ymin,  
    unsigned int ymax,  
    unsigned int zmin,  
    unsigned int zmax,  
    std::istream & is) [protected]
```

12.175.3.10 GetBufferLength()

unsigned long gdcm::JPEGLSCodec::GetBufferLength () const [inline]

12.175.3.11 GetHeaderInfo()

bool gdcm::JPEGLSCodec::GetHeaderInfo (
std::istream & is,
[TransferSyntax](#) & ts) [override], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.12 GetLossless()

bool gdcm::JPEGLSCodec::GetLossless () const

12.175.3.13 IsFrameEncoder()

bool gdcm::JPEGLSCodec::IsFrameEncoder () [override], [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.14 IsRowEncoder()

bool gdcm::JPEGLSCodec::IsRowEncoder () [override], [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.15 SetBufferLength()

void gdcm::JPEGLSCodec::SetBufferLength (
unsigned long l) [inline]

12.175.3.16 SetLossless()

void gdcm::JPEGLSCodec::SetLossless (
bool l)

12.175.3.17 SetLossyError()

void gdcm::JPEGLSCodec::SetLossyError (
int error)

[0-3] generally

12.175.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.175.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.175.4 Friends And Related Symbol Documentation

12.175.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

References [ImageRegionReader](#).

Referenced by [ImageRegionReader](#).

The documentation for this class was generated from the following file:

- [gdcm.JPEGLSCodec.h](#)

12.176 gdcm::JSON Class Reference

```
#include <gdcmJSON.h>
```

Public Member Functions

- [JSON](#) ()
- [~JSON](#) ()
- bool [Code](#) ([DataSet](#) const &in, std::ostream &os)
- bool [Decode](#) (std::istream &is, [DataSet](#) &out)
- bool [GetPrettyPrint](#) () const
- void [PrettyPrintOff](#) ()
- void [PrettyPrintOn](#) ()
- void [SetPrettyPrint](#) (bool onoff)

12.176.1 Detailed Description

Examples

[QIDO-RS.cxx](#).

12.176.2 Constructor & Destructor Documentation

12.176.2.1 JSON()

```
gdcm::JSON::JSON ()
```

12.176.2.2 ~JSON()

```
gdcm::JSON::~~JSON ()
```

12.176.3 Member Function Documentation

12.176.3.1 Code()

```
bool gdcm::JSON::Code (  
    DataSet const & in,  
    std::ostream & os)
```

Examples

[QIDO-RS.cxx](#).

12.176.3.2 Decode()

```
bool gdcm::JSON::Decode (  
    std::istream & is,  
    DataSet & out)
```

Examples

[QIDO-RS.cxx](#).

12.176.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint () const
```

12.176.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ()
```

12.176.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ()
```

Examples

[QIDO-RS.cxx](#).

12.176.3.6 SetPrettyPrint()

```
void gdcm::JSON::SetPrettyPrint (  
    bool onoff)
```

The documentation for this class was generated from the following file:

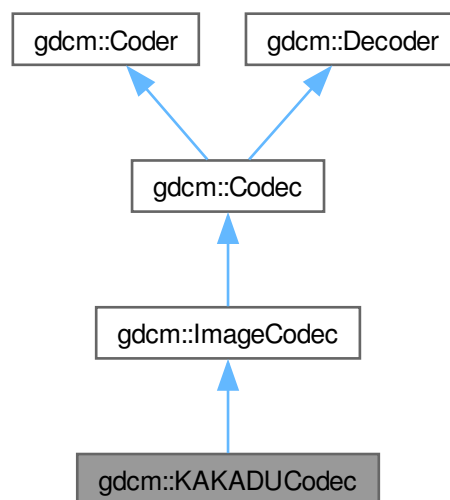
- [gdcm.JSON.h](#)

12.177 gdcm::KAKADUCodec Class Reference

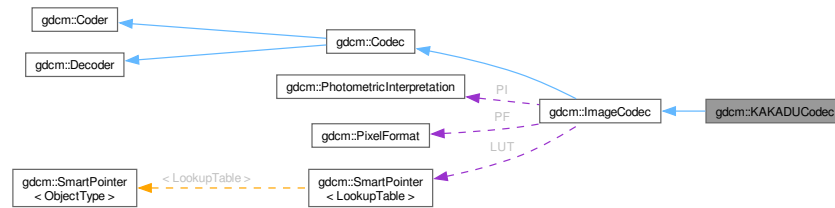
[KAKADUCodec](#).

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for `gdcm::KAKADUCodec`:



Public Member Functions

- [KAKADUCodec](#) ()
- [~KAKADUCodec](#) () override
- [bool CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- [bool CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec * Clone](#) () const override
- [bool Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Code.
- [bool Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- [bool CleanupUnusedBits](#) (char *data, size_t datalen)
- [const unsigned int * GetDimensions](#) () const
- [virtual bool GetHeaderInfo](#) (std::istream &is_, [TransferSyntax](#) &ts)
- [bool GetLossyFlag](#) () const
- [const LookupTable & GetLUT](#) () const
- [bool GetNeedByteSwap](#) () const
- [unsigned int GetNumberOfDimensions](#) () const
- [const PhotometricInterpretation & GetPhotometricInterpretation](#) () const
- [PixelFormat & GetPixelFormat](#) ()
- [const PixelFormat & GetPixelFormat](#) () const
- [unsigned int GetPlanarConfiguration](#) () const
- [bool IsLossy](#) () const
- [void SetDimensions](#) (const std::vector< unsigned int > &d)
- [void SetDimensions](#) (const unsigned int d[3])
- [void SetLossyFlag](#) (bool l)
- [void SetLUT](#) ([LookupTable](#) const &lut)
- [void SetNeedByteSwap](#) (bool b)
- [void SetNeedOverlayCleanup](#) (bool b)
- [void SetNumberOfDimensions](#) (unsigned int dim)
- [void SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- [virtual void SetPixelFormat](#) ([PixelFormat](#) const &pf)
- [void SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.177.1 Detailed Description

[KAKADUCodec](#).

12.177.2 Constructor & Destructor Documentation

12.177.2.1 KAKADUCodec()

`gdcm::KAKADUCodec::KAKADUCodec ()`

12.177.2.2 ~KAKADUCodec()

`gdcm::KAKADUCodec::~~KAKADUCodec ()` [override]

12.177.3 Member Function Documentation

12.177.3.1 CanCode()

`bool gdcm::KAKADUCodec::CanCode (`
 [TransferSyntax](#) const &) const [override], [virtual]

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.177.3.2 CanDecode()

`bool gdcm::KAKADUCodec::CanDecode (`
 [TransferSyntax](#) const &) const [override], [virtual]

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.177.3.3 Clone()

`ImageCodec * gdcm::KAKADUCodec::Clone ()` const [override], [virtual]

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.177.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
    DataElement const & in_,
    DataElement & out_) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

12.177.3.5 Decode()

```
bool gdcm::KAKADUCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

The documentation for this class was generated from the following file:

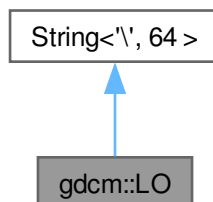
- [gdcmKAKADUCodec.h](#)

12.178 gdcm::LO Class Reference

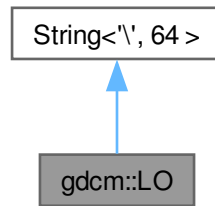
[LO](#).

```
#include <gdcmLO.h>
```

Inheritance diagram for gdcm::LO:



Collaboration diagram for `gdcm::LO`:



Public Types

- typedef `Superclass::const_iterator` `const_iterator`
- typedef `Superclass::const_reference` `const_reference`
- typedef `Superclass::const_reverse_iterator` `const_reverse_iterator`
- typedef `Superclass::difference_type` `difference_type`
- typedef `Superclass::iterator` `iterator`
- typedef `Superclass::pointer` `pointer`
- typedef `Superclass::reference` `reference`
- typedef `Superclass::reverse_iterator` `reverse_iterator`
- typedef `Superclass::size_type` `size_type`
- typedef `String<'\\', 64 > Superclass`
- typedef `Superclass::value_type` `value_type`

Public Member Functions

- `LO ()`
- `LO (const Superclass &s, size_type pos=0, size_type n=npos)`
- `LO (const value_type *s)`
- `LO (const value_type *s, size_type n)`
- `bool IsValid () const`

12.178.1 Detailed Description

`LO`.

Note

TODO

12.178.2 Member Typedef Documentation

12.178.2.1 `const_iterator`

typedef [Superclass::const_iterator](#) [gdcm::LO::const_iterator](#)

12.178.2.2 `const_reference`

typedef [Superclass::const_reference](#) [gdcm::LO::const_reference](#)

12.178.2.3 `const_reverse_iterator`

typedef [Superclass::const_reverse_iterator](#) [gdcm::LO::const_reverse_iterator](#)

12.178.2.4 `difference_type`

typedef [Superclass::difference_type](#) [gdcm::LO::difference_type](#)

12.178.2.5 `iterator`

typedef [Superclass::iterator](#) [gdcm::LO::iterator](#)

12.178.2.6 `pointer`

typedef [Superclass::pointer](#) [gdcm::LO::pointer](#)

12.178.2.7 `reference`

typedef [Superclass::reference](#) [gdcm::LO::reference](#)

12.178.2.8 `reverse_iterator`

typedef [Superclass::reverse_iterator](#) [gdcm::LO::reverse_iterator](#)

12.178.2.9 `size_type`

typedef [Superclass::size_type](#) [gdcm::LO::size_type](#)

12.178.2.10 Superclass

```
typedef String<'\\',64> gdcm::LO::Superclass
```

12.178.2.11 value_type

```
typedef Superclass::value\_type gdcm::LO::value\_type
```

12.178.3 Constructor & Destructor Documentation

12.178.3.1 LO() [1/4]

```
gdcm::LO::LO () [inline]
```

12.178.3.2 LO() [2/4]

```
gdcm::LO::LO (  
    const value\_type * s) [inline]
```

12.178.3.3 LO() [3/4]

```
gdcm::LO::LO (  
    const value\_type * s,  
    size\_type n) [inline]
```

12.178.3.4 LO() [4/4]

```
gdcm::LO::LO (  
    const Superclass & s,  
    size\_type pos = 0,  
    size\_type n = npos) [inline]
```

12.178.4 Member Function Documentation

12.178.4.1 IsValid()

```
bool gdcm::LO::IsValid () const [inline]
```

References [gdcm::String<'\\', 64 >::IsValid\(\)](#).

The documentation for this class was generated from the following file:

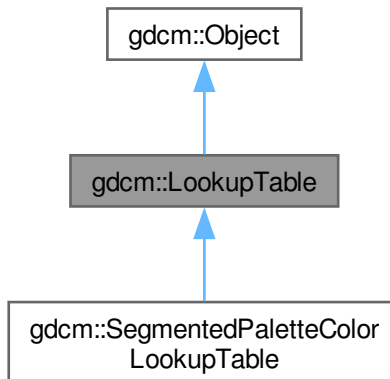
- [gdcmLO.h](#)

12.179 gdcm::LookupTable Class Reference

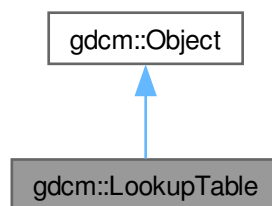
[LookupTable](#) class.

```
#include <gdcmLookupTable.h>
```

Inheritance diagram for gdcm::LookupTable:



Collaboration diagram for gdcm::LookupTable:



Public Types

- enum [LookupTableType](#) {
 [RED](#) = 0 ,
 [GREEN](#) ,
 [BLUE](#) ,
 [GRAY](#) ,
 [UNKNOWN](#) }

Public Member Functions

- [LookupTable](#) ()
- [LookupTable](#) ([LookupTable](#) const &lut)
- [~LookupTable](#) () override
- void [Allocate](#) (unsigned short bitsample=8)
Allocate the LUT.
- void [Clear](#) ()
Clear the LUT.
- bool [Decode](#) (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- void [Decode](#) (std::istream &is, std::ostream &os) const
Decode the LUT.
- bool [Decode8](#) (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
Decode into RGB 8 bits space.
- unsigned short [GetBitSample](#) () const
return the bit sample
- bool [GetBufferAsRGBA](#) (unsigned char *rgba) const
return the LUT as RGBA buffer
- void [GetLUT](#) ([LookupTableType](#) type, unsigned char *array, unsigned int &length) const
- void [GetLUTDescriptor](#) ([LookupTableType](#) type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int [GetLUTLength](#) ([LookupTableType](#) type) const
- const unsigned char * [GetPointer](#) () const
return a raw pointer to the LUT
- void [InitializeBlueLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool [Initialized](#) () const
return whether the LUT has been initialized
- void [InitializeGreenLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void [InitializeLUT](#) ([LookupTableType](#) type, unsigned short length, unsigned short subscript, unsigned short bitsize)
Generic interface:
- void [InitializeRedLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
RED / GREEN / BLUE specific:
- bool [IsRGB8](#) () const
Return whether 16 bits LUT is in RGB 8 bits space.
- void [Print](#) (std::ostream &) const override
- void [SetBlueLUT](#) (const unsigned char *blue, unsigned int length)
- void [SetGreenLUT](#) (const unsigned char *green, unsigned int length)
- virtual void [SetLUT](#) ([LookupTableType](#) type, const unsigned char *array, unsigned int length)
- void [SetRedLUT](#) (const unsigned char *red, unsigned int length)
- bool [WriteBufferAsRGBA](#) (const unsigned char *rgba)
Write the LUT as RGBA.

Public Member Functions inherited from [gdcem::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Protected Attributes

- unsigned short [BitSample](#)
- bool [IncompleteLUT:1](#)
- LookupTableInternal * [Internal](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.179.1 Detailed Description

[LookupTable](#) class.

Examples

[ExtractImageRegionWithLUT.cs](#), and [PrintLUT.cxx](#).

12.179.2 Member Enumeration Documentation

12.179.2.1 LookupTableType

enum [gdcm::LookupTable::LookupTableType](#)

Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

12.179.3 Constructor & Destructor Documentation

12.179.3.1 LookupTable() [1/2]

[gdcm::LookupTable::LookupTable](#) ()

Referenced by [LookupTable\(\)](#).

12.179.3.2 ~LookupTable()

```
gdcm::LookupTable::~~LookupTable () [override]
```

12.179.3.3 LookupTable() [2/2]

```
gdcm::LookupTable::LookupTable (
    LookupTable const & lut) [inline]
```

References [LookupTable\(\)](#), [gdcm::Object::Object\(\)](#), [BitSample](#), [gdcm_assert](#), [IncompleteLUT](#), and [Internal](#).

12.179.4 Member Function Documentation

12.179.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
    unsigned short bitsample = 8)
```

Allocate the LUT.

12.179.4.2 Clear()

```
void gdcm::LookupTable::Clear ()
```

Clear the LUT.

12.179.4.3 Decode() [1/2]

```
bool gdcm::LookupTable::Decode (
    char * outputbuffer,
    size_t outlen,
    const char * inputbuffer,
    size_t inlen) const
```

Decode the LUT outputbuffer will contains the RGB decoded PALETTE COLOR input image of size inlen the outputbuffer should be at least 3 times the size of inlen

12.179.4.4 Decode() [2/2]

```
void gdcm::LookupTable::Decode (
    std::istream & is,
    std::ostream & os) const
```

Decode the LUT.

Examples

[ExtractImageRegionWithLUT.cs](#).

12.179.4.5 Decode8()

```
bool gdcm::LookupTable::Decode8 (
    char * outputbuffer,
    size_t outlen,
    const char * inputbuffer,
    size_t inlen) const
```

Decode into RGB 8 bits space.

12.179.4.6 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample () const [inline]
```

return the bit sample

References [BitSample](#).

12.179.4.7 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
    unsigned char * rgba) const
```

return the LUT as RGBA buffer

12.179.4.8 GetLUT()

```
void gdcm::LookupTable::GetLUT (
    LookupTableType type,
    unsigned char * array,
    unsigned int & length) const
```

12.179.4.9 GetLUTDescriptor()

```
void gdcm::LookupTable::GetLUTDescriptor (
    LookupTableType type,
    unsigned short & length,
    unsigned short & subscript,
    unsigned short & bitsize) const
```

12.179.4.10 GetLUTLength()

```
unsigned int gdcm::LookupTable::GetLUTLength (
    LookupTableType type) const
```

12.179.4.11 GetPointer()

```
const unsigned char * gdcm::LookupTable::GetPointer () const
```

return a raw pointer to the LUT

12.179.4.12 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize)
```

12.179.4.13 Initialized()

```
bool gdcm::LookupTable::Initialized () const
```

return whether the LUT has been initialized

12.179.4.14 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize)
```

12.179.4.15 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
    LookupTableType type,
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize)
```

Generic interface:

12.179.4.16 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize)
```

RED / GREEN / BLUE specific:

12.179.4.17 IsRGB8()

```
bool gdcm::LookupTable::IsRGB8 () const
```

Return whether 16 bits LUT is in RGB 8 bits space.

12.179.4.18 Print()

```
void gdcm::LookupTable::Print (  
    std::ostream & ) const    [override], [virtual]
```

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

Examples

[PrintLUT.cxx](#).

12.179.4.19 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (  
    const unsigned char * blue,  
    unsigned int length)
```

12.179.4.20 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (  
    const unsigned char * green,  
    unsigned int length)
```

12.179.4.21 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (  
    LookupTableType type,  
    const unsigned char * array,  
    unsigned int length)    [virtual]
```

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

12.179.4.22 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (  
    const unsigned char * red,  
    unsigned int length)
```

12.179.4.23 WriteBufferAsRGBA()

```
bool gdcM::LookupTable::WriteBufferAsRGBA (  
    const unsigned char * rgba)
```

Write the LUT as RGBA.

12.179.5 Member Data Documentation

12.179.5.1 BitSample

```
unsigned short gdcM::LookupTable::BitSample [protected]
```

Referenced by [LookupTable\(\)](#), and [GetBitSample\(\)](#).

12.179.5.2 IncompleteLUT

```
bool gdcM::LookupTable::IncompleteLUT [protected]
```

Referenced by [LookupTable\(\)](#).

12.179.5.3 Internal

```
LookupTableInternal* gdcM::LookupTable::Internal [protected]
```

Referenced by [LookupTable\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMLookupTable.h](#)

12.180 gdcM::Scanner2::ltstr Struct Reference

```
#include <gdcMScanner2.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char *s1, const char *s2) const

12.180.1 Member Function Documentation

12.180.1.1 operator>()()

```
bool gdcm::Scanner2::ltstr::operator() (  
    const char * s1,  
    const char * s2) const    [inline]
```

References [gdcm_assert](#).

The documentation for this struct was generated from the following file:

- [gdcmScanner2.h](#)

12.181 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char *s1, const char *s2) const

12.181.1 Member Function Documentation

12.181.1.1 operator>()()

```
bool gdcm::Scanner::ltstr::operator() (  
    const char * s1,  
    const char * s2) const    [inline]
```

References [gdcm_assert](#).

The documentation for this struct was generated from the following file:

- [gdcmScanner.h](#)

12.182 gdcm::StrictScanner2::ltstr Struct Reference

```
#include <gdcmStrictScanner2.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char *s1, const char *s2) const

12.182.1 Member Function Documentation

12.182.1.1 [operator\(\)](#)()

```
bool gdcm::StrictScanner2::ltstr::operator() (  
    const char * s1,  
    const char * s2) const    [inline]
```

References [gdcm_assert](#).

The documentation for this struct was generated from the following file:

- [gdcmStrictScanner2.h](#)

12.183 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char *s1, const char *s2) const

12.183.1 Member Function Documentation

12.183.1.1 [operator\(\)](#)()

```
bool gdcm::StrictScanner::ltstr::operator() (  
    const char * s1,  
    const char * s2) const    [inline]
```

References [gdcm_assert](#).

The documentation for this struct was generated from the following file:

- [gdcmStrictScanner.h](#)

12.184 gdcmmacro::Macro Class Reference

Class for representing a [Macro](#).

```
#include <gdcmmacro.h>
```

Public Types

- typedef std::vector< std::string > [ArrayIncludeMacrosType](#)
- typedef std::map< [Tag](#), [MacroEntry](#) > [MapModuleEntry](#)

Public Member Functions

- [Macro](#) ()=default
- void [AddMacroEntry](#) (const [Tag](#) &tag, const [MacroEntry](#) &module)
Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.
- void [Clear](#) ()
- bool [FindMacroEntry](#) (const [Tag](#) &tag) const
- const [MacroEntry](#) & [GetMacroEntry](#) (const [Tag](#) &tag) const
- const char * [GetName](#) () const
- void [SetName](#) (const char *name)
- bool [Verify](#) (const [DataSet](#) &ds, [Usage](#) const &usage) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Macro](#) &_val)

12.184.1 Detailed Description

Class for representing a [Macro](#).

Note

[Attribute Macro](#): a set of Attributes that are described in a single table that is referenced by multiple [Module](#) or other tables.

See also

[Module](#)

12.184.2 Member Typedef Documentation

12.184.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcmmacro::Macro::ArrayIncludeMacrosType
```

12.184.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcmmacro::Macro::MapModuleEntry
```

12.184.3 Constructor & Destructor Documentation

12.184.3.1 Macro()

```
gdcmmacro::Macro::Macro () [default]
```

References [Macro\(\)](#), and [operator<<](#).

Referenced by [Macro\(\)](#), and [operator<<](#).

12.184.4 Member Function Documentation

12.184.4.1 AddMacroEntry()

```
void gdcmmacro::Macro::AddMacroEntry (
    const Tag & tag,
    const MacroEntry & module) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

12.184.4.2 Clear()

```
void gdcmmacro::Macro::Clear () [inline]
```

12.184.4.3 FindMacroEntry()

```
bool gdcmmacro::Macro::FindMacroEntry (
    const Tag & tag) const
```

Find or Get a [ModuleEntry](#). [ModuleEntry](#) are either search are root-level or within nested-macro included in module.

12.184.4.4 GetMacroEntry()

```
const MacroEntry & gdcmmacro::Macro::GetMacroEntry (
    const Tag & tag) const
```

12.184.4.5 GetName()

```
const char * gdcmmacro::Macro::GetName () const [inline]
```

12.184.4.6 SetName()

```
void gdcmmacro::Macro::SetName (
    const char * name) [inline]
```

12.184.4.7 Verify()

```
bool gdcmmacro::Macro::Verify (
    const DataSet & ds,
    Usage const & usage) const
```

12.184.5 Friends And Related Symbol Documentation

12.184.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Macro & _val) [friend]
```

References [Macro\(\)](#).

Referenced by [Macro\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmacro.h](#)

12.185 gdcmmacro::Macro Class Reference

Class for representing a [Module](#).

```
#include <gdcmmacro.h>
```

Public Types

- typedef std::map< std::string, [Macro](#) > [ModuleMapType](#)

Public Member Functions

- [Macro](#) ()=default
- void [AddMacro](#) (const char *ref, const [Macro](#) &module)
- void [Clear](#) ()
- const [Macro](#) & [GetMacro](#) (const char *name) const
- bool [IsEmpty](#) () const

Friends

- `std::ostream & operator<< (std::ostream &_os, const Macros &_val)`

12.185.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples

[TraverseModules.cxx](#).

12.185.2 Member Typedef Documentation

12.185.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType
```

12.185.3 Constructor & Destructor Documentation

12.185.3.1 Macros()

```
gdcm::Macros::Macros () [default]
```

References [Macros\(\)](#), and [operator<<](#).

Referenced by [Macros\(\)](#), and [operator<<](#).

12.185.4 Member Function Documentation

12.185.4.1 AddMacro()

```
void gdcm::Macros::AddMacro (  
    const char * ref,  
    const Macro & module) [inline]
```

References [gdcm_assert](#).

12.185.4.2 Clear()

```
void gdcm::Macros::Clear () [inline]
```

12.185.4.3 GetMacro()

```
const Macro & gdcm::Macros::GetMacro (
    const char * name) const [inline]
```

References [gdcm_assert](#).

12.185.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty () const [inline]
```

12.185.5 Friends And Related Symbol Documentation

12.185.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const Macros & __val) [friend]
```

References [Macros\(\)](#).

Referenced by [Macros\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmMacros.h](#)

12.186 gdcm::network::MaximumLengthSub Class Reference

[MaximumLengthSub](#).

```
#include <gdcmMaximumLengthSub.h>
```

Public Member Functions

- [MaximumLengthSub](#) ()
- uint32_t [GetMaximumLength](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetMaximumLength](#) (uint32_t maximumlength)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.186.1 Detailed Description

[MaximumLengthSub](#).

Annex D [Table D.1-1](#) MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

or

[Table D.1-2](#) Maximum length sub-item fields (A-ASSOCIATE-AC)

12.186.2 Constructor & Destructor Documentation

12.186.2.1 MaximumLengthSub()

```
gdcm::network::MaximumLengthSub::MaximumLengthSub ()
```

12.186.3 Member Function Documentation

12.186.3.1 GetMaximumLength()

```
uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength () const [inline]
```

12.186.3.2 Print()

```
void gdcm::network::MaximumLengthSub::Print (  
    std::ostream & os) const
```

12.186.3.3 Read()

```
std::istream & gdcm::network::MaximumLengthSub::Read (  
    std::istream & is)
```

12.186.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (  
    uint32_t maximumlength)
```

12.186.3.5 Size()

```
size_t gdcm::network::MaximumLengthSub::Size () const
```

12.186.3.6 Write()

```
const std::ostream & gdcm::network::MaximumLengthSub::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmMaximumLengthSub.h](#)

12.187 gdcm::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcmMD5.h>
```

Static Public Member Functions

- static bool [Compute](#) (const char *buffer, size_t buf_len, char digest_str[33])
- static bool [ComputeFile](#) (const char *filename, char digest_str[33])
Compute md5 from a file filename.

12.187.1 Detailed Description

Class for [MD5](#).

Warning

this class is able to pick from two implementations:

1. a lightweight md5 implementation (when GDCM_BUILD_TESTING is turned ON)
2. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

12.187.2 Member Function Documentation

12.187.2.1 Compute()

```
bool gdcm::MD5::Compute (  
    const char * buffer,  
    size_t buf_len,  
    char digest_str[33]) [static]
```

12.187.2.2 ComputeFile()

```
bool gdcM::MD5::ComputeFile (
    const char * filename,
    char digest_str[33]) [static]
```

Compute md5 from a file filename.

The documentation for this class was generated from the following file:

- [gdcMMD5.h](#)

12.188 gdcM::MEC_MR3 Class Reference

Class for [MEC_MR3](#).

```
#include <gdcMMEC_MR3.h>
```

Static Public Member Functions

- static const [PrivateTag](#) & [GetCanonMECMR3Tag](#) ()
- static const [PrivateTag](#) & [GetPMTFInformationDataTag](#) ()
- static const [PrivateTag](#) & [GetToshibaMECMR3Tag](#) ()
- static bool [Print](#) (const char *src, size_t srclen)

12.188.1 Detailed Description

Class for [MEC_MR3](#).

12.188.2 Member Function Documentation

12.188.2.1 GetCanonMECMR3Tag()

```
const PrivateTag & gdcM::MEC_MR3::GetCanonMECMR3Tag () [static]
```

Return the private tag used by CANON to store the [MEC_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"↵CANON_MEC_MR3");

12.188.2.2 GetPMTFInformationDataTag()

```
const PrivateTag & gdcM::MEC_MR3::GetPMTFInformationDataTag () [static]
```

Return the private tag used by PMTF to store the [MEC_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"PMTF INFORMATION DATA");

12.188.2.3 GetToshibaMECMR3Tag()

```
const PrivateTag & gdcm::MEC_MR3::GetToshibaMECMR3Tag () [static]
```

Return the private tag used by TOSHIBA to store the [MEC_MR3](#) data This is: [PrivateTag](#)(0x0029,0x90,"↵TOSHIBA_MEC_MR3");

12.188.2.4 Print()

```
bool gdcm::MEC_MR3::Print (
    const char * src,
    size_t srclen) [static]
```

The documentation for this class was generated from the following file:

- [gdcmMEC_MR3.h](#)

12.189 gdcm::MediaStorage Class Reference

[MediaStorage](#).

```
#include <gdcmMediaStorage.h>
```

Public Types

- enum [MSType](#) {
[MediaStorageDirectoryStorage](#) = 0 ,
[ComputedRadiographyImageStorage](#) ,
[DigitalXRayImageStorageForPresentation](#) ,
[DigitalXRayImageStorageForProcessing](#) ,
[DigitalMammographyImageStorageForPresentation](#) ,
[DigitalMammographyImageStorageForProcessing](#) ,
[DigitalIntraoralXrayImageStorageForPresentation](#) ,
[DigitalIntraoralXrayImageStorageForProcessing](#) ,
[CTImageStorage](#) ,
[EnhancedCTImageStorage](#) ,
[UltrasoundImageStorageRetired](#) ,
[UltrasoundImageStorage](#) ,
[UltrasoundMultiFrameImageStorageRetired](#) ,
[UltrasoundMultiFrameImageStorage](#) ,
[MRImageStorage](#) ,
[EnhancedMRImageStorage](#) ,
[MRSpectroscopyStorage](#) ,
[NuclearMedicineImageStorageRetired](#) ,
[SecondaryCaptureImageStorage](#) ,
[MultiframeSingleBitSecondaryCaptureImageStorage](#) ,
[MultiframeGrayscaleByteSecondaryCaptureImageStorage](#) ,
[MultiframeGrayscaleWordSecondaryCaptureImageStorage](#) ,

[MultiframeTrueColorSecondaryCaptureImageStorage](#) ,
[StandaloneOverlayStorage](#) ,
[StandaloneCurveStorage](#) ,
[LeadECGWaveformStorage](#) ,
[GeneralECGWaveformStorage](#) ,
[AmbulatoryECGWaveformStorage](#) ,
[HemodynamicWaveformStorage](#) ,
[CardiacElectrophysiologyWaveformStorage](#) ,
[BasicVoiceAudioWaveformStorage](#) ,
[StandaloneModalityLUTStorage](#) ,
[StandaloneVOILUTStorage](#) ,
[GrayscaleSoftcopyPresentationStateStorageSOPClass](#) ,
[XRayAngiographicImageStorage](#) ,
[XRayRadiofluoroscopicImageStorage](#) ,
[XRayAngiographicBiPlaneImageStorageRetired](#) ,
[NuclearMedicineImageStorage](#) ,
[RawDataStorage](#) ,
[SpacialRegistrationStorage](#) ,
[SpacialFiducialsStorage](#) ,
[PETImageStorage](#) ,
[RTImageStorage](#) ,
[RTDoseStorage](#) ,
[RTStructureSetStorage](#) ,
[RTPlanStorage](#) ,
[CSANonImageStorage](#) ,
[Philips3D](#) ,
[EnhancedSR](#) ,
[BasicTextSR](#) ,
[HardcopyGrayscaleImageStorage](#) ,
[ComprehensiveSR](#) ,
[DetachedStudyManagementSOPClass](#) ,
[EncapsulatedPDFStorage](#) ,
[EncapsulatedCDASStorage](#) ,
[StudyComponentManagementSOPClass](#) ,
[DetachedVisitManagementSOPClass](#) ,
[DetachedPatientManagementSOPClass](#) ,
[VideoEndoscopicImageStorage](#) ,
[GeneralElectricMagneticResonanceImageStorage](#) ,
[GEPrivate3DModelStorage](#) ,
[ToshibaPrivateDataStorage](#) ,
[MammographyCADSR](#) ,
[KeyObjectSelectionDocument](#) ,
[HangingProtocolStorage](#) ,
[ModalityPerformedProcedureStepSOPClass](#) ,
[PhilipsPrivateMRSyntheticImageStorage](#) ,
[VLPhotographicImageStorage](#) ,
[SegmentationStorage](#) ,
[RTIonPlanStorage](#) ,
[XRay3DAngiographicImageStorage](#) ,
[EnhancedXAImageStorage](#) ,
[RTIonBeamsTreatmentRecordStorage](#) ,
[SurfaceSegmentationStorage](#) ,
[VLWholeSlideMicroscopyImageStorage](#) ,
[RTTreatmentSummaryRecordStorage](#) ,

- EnhancedUSVolumeStorage ,
- XRayRadiationDoseSR ,
- VLEndoscopicImageStorage ,
- BreastTomosynthesisImageStorage ,
- FujiPrivateCRImageStorage ,
- OphthalmicPhotography8BitImageStorage ,
- OphthalmicTomographyImageStorage ,
- VLMicroscopicImageStorage ,
- EnhancedPETImageStorage ,
- VideoPhotographicImageStorage ,
- XRay3DCraniofacialImageStorage ,
- IVOCTForPresentation ,
- IVOCTForProcessing ,
- LegacyConvertedEnhancedCTImageStorage ,
- LegacyConvertedEnhancedMRImageStorage ,
- LegacyConvertedEnhancedPETImageStorage ,
- BreastProjectionXRayImageStorageForPresentation ,
- BreastProjectionXRayImageStorageForProcessing ,
- HardcopyColorImageStorage ,
- EnhancedMRColorImageStorage ,
- FujiPrivateMammoCRImageStorage ,
- OphthalmicPhotography16BitImageStorage ,
- VideoMicroscopicImageStorage ,
- MS_END }
- enum ObjectType {
 - NoObject = 0 ,
 - Video ,
 - Waveform ,
 - Audio ,
 - PDF ,
 - URI ,
 - Segmentation ,
 - ObjectEnd }

Public Member Functions

- MediaStorage (MSType type=MS_END)
- const char * GetModality () const
- unsigned int GetModalityDimension () const
- const char * GetString () const
 - Return the Media String of the object.
- void GuessFromModality (const char *modality, unsigned int dimension=2)
- bool IsUndefined () const
- operator MSType () const
- bool SetFromDataSet (DataSet const &ds)
- bool SetFromFile (File const &file)
- bool SetFromHeader (FileMetaInformation const &fmi)
- bool SetFromModality (DataSet const &ds)

Static Public Member Functions

- static const char * [GetMSString](#) (MSType ts)
Return the Media [String](#) associated. Will return NULL for MS_END.
- static MSType [GetMSType](#) (const char *str)
- static unsigned int [GetNumberOfModality](#) ()
- static unsigned int [GetNumberOfMSString](#) ()
- static unsigned int [GetNumberOfMSType](#) ()
- static bool [IsImage](#) (MSType ts)

Protected Member Functions

- void [SetFromSourceImageSequence](#) (DataSet const &ds)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [MediaStorage](#) &ms)

12.189.1 Detailed Description

[MediaStorage](#).

Note

FIXME There should not be any notion of [Image](#) and/or PDF at that point Only the codec can answer yes I support this Media Storage or not... For instance an [ImageCodec](#) will answer yes to most of them while a [PDFCodec](#) will answer only for the Encapsulated PDF

See also

[UIDs](#)

Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), and [iU22tomultisc.cxx](#).

12.189.2 Member Enumeration Documentation

12.189.2.1 MStype

enum [gdcmm::MediaStorage::MStype](#)

Enumerator

MediaStorageDirectoryStorage	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyImageStorageForPresentation	
DigitalMammographyImageStorageForProcessing	
DigitalIntraoralXRayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
UltrasoundMultiFrameImageStorageRetired	
UltrasoundMultiFrameImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorage	
StandaloneCurveStorage	
LeadECGWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorage	
StandaloneVOILUTStorage	

GrayscaleSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
XRayRadiofluoroscopicImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTPlanStorage	
CSANonImageStorage	
Philips3D	
EnhancedSR	
BasicTextSR	
HardcopyGrayscaleImageStorage	
ComprehensiveSR	
DetachedStudyManagementSOPClass	
EncapsulatedPDFStorage	
EncapsulatedCDASStorage	
StudyComponentManagementSOPClass	
DetachedVisitManagementSOPClass	
DetachedPatientManagementSOPClass	
VideoEndoscopicImageStorage	
GeneralElectricMagneticResonanceImageStorage	
GEPrivate3DModelStorage	
ToshibaPrivateDataStorage	
MammographyCADSR	
KeyObjectSelectionDocument	
HangingProtocolStorage	
ModalityPerformedProcedureStepSOPClass	
PhilipsPrivateMRSyntheticImageStorage	
VLPhotographicImageStorage	
SegmentationStorage	
RTIonPlanStorage	
XRay3DAngiographicImageStorage	
EnhancedXAImageStorage	

RTIonBeamsTreatmentRecordStorage	
SurfaceSegmentationStorage	
VLWholeSlideMicroscopyImageStorage	
RTTreatmentSummaryRecordStorage	
EnhancedUSVolumeStorage	
XRayRadiationDoseSR	
VLEndoscopicImageStorage	
BreastTomosynthesisImageStorage	
FujiPrivateCRImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicTomographyImageStorage	
VLMicroscopicImageStorage	
EnhancedPETImageStorage	
VideoPhotographicImageStorage	
XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRIImageStorage	
LegacyConvertedEnhancedPETImageStorage	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
HardcopyColorImageStorage	
EnhancedMRColorImageStorage	
FujiPrivateMammoCRImageStorage	
OphthalmicPhotography16BitImageStorage	
VideoMicroscopicImageStorage	
MS_END	

Examples

[GenerateStandardSOPClasses.cxx](#), and [MpegVideoInfo.cs](#).

12.189.2.2 ObjectType

enum [gdcmm::MediaStorage::ObjectType](#)

Enumerator

NoObject	
----------	--

Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

12.189.3 Constructor & Destructor Documentation

12.189.3.1 MediaStorage()

gdcm::MediaStorage::MediaStorage (
[MSType](#) type = [MS_END](#)) [inline]

References [MS_END](#).

Referenced by [GuessFromModality\(\)](#), and [operator<<](#).

12.189.4 Member Function Documentation

12.189.4.1 GetModality()

const char * gdcm::MediaStorage::GetModality () const

12.189.4.2 GetModalityDimension()

unsigned int gdcm::MediaStorage::GetModalityDimension () const

12.189.4.3 GetMSString()

const char * gdcm::MediaStorage::GetMSString (
[MSType](#) ts) [static]

Return the Media [String](#) associated. Will return NULL for [MS_END](#).

Examples

[GenerateStandardSOPClasses.cxx](#).

Referenced by [operator<<](#).

12.189.4.4 GetMSType()

[MSType](#) gdcmm::MediaStorage::GetMSType (
 const char * str) [static]

Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

12.189.4.5 GetNumberOfModality()

unsigned int gdcmm::MediaStorage::GetNumberOfModality () [static]

12.189.4.6 GetNumberOfMSString()

unsigned int gdcmm::MediaStorage::GetNumberOfMSString () [static]

12.189.4.7 GetNumberOfMSType()

unsigned int gdcmm::MediaStorage::GetNumberOfMSType () [static]

12.189.4.8 GetString()

const char * gdcmm::MediaStorage::GetString () const

Return the Media [String](#) of the object.

Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#),
[GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#),
and [iU22tomultisc.cxx](#).

12.189.4.9 GuessFromModality()

void gdcmm::MediaStorage::GuessFromModality (
 const char * modality,
 unsigned int dimension = 2)

References [MediaStorage\(\)](#), and [operator<<](#).

12.189.4.10 IsImage()

```
bool gdcm::MediaStorage::IsImage (
    MSType ts) [static]
```

Returns whether DICOM has a Pixel Data element (7fe0,0010)

Warning

[MRSpectroscopyStorage](#) could be image but are not

Examples

[MetaImageMD5Activiz.cs](#).

12.189.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined () const [inline]
```

Examples

[TestReader.cxx](#).

References [MS_END](#).

12.189.4.12 operator MSType()

```
gdcm::MediaStorage::operator MSType () const [inline]
```

12.189.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
    DataSet const & ds)
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

12.189.4.14 SetFromFile()

```
bool gdcm::MediaStorage::SetFromFile (
    File const & file)
```

Attempt to set the [MediaStorage](#) from a file: WARNING: When no [MediaStorage](#) & Modality are found BUT a PixelData element is found then [MediaStorage](#) is set to the default [SecondaryCaptureImageStorage](#) (return value is false in this case)

Examples

[ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [TestReader.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

12.189.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
    FileMetaInformation const & fmi)
```

12.189.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
    DataSet const & ds)
```

12.189.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
    DataSet const & ds) [protected]
```

12.189.5 Friends And Related Symbol Documentation

12.189.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const MediaStorage & ms) [friend]
```

References [MediaStorage\(\)](#), [GetMSSString\(\)](#), and [operator<<](#).

Referenced by [GuessFromModality\(\)](#), and [operator<<](#).

The documentation for this class was generated from the following file:

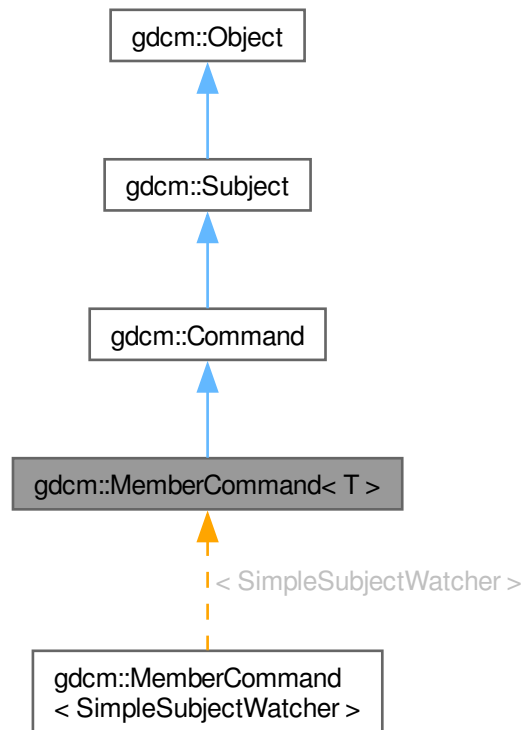
- [gdcmMediaStorage.h](#)

12.190 gdcm::MemberCommand< T > Class Template Reference

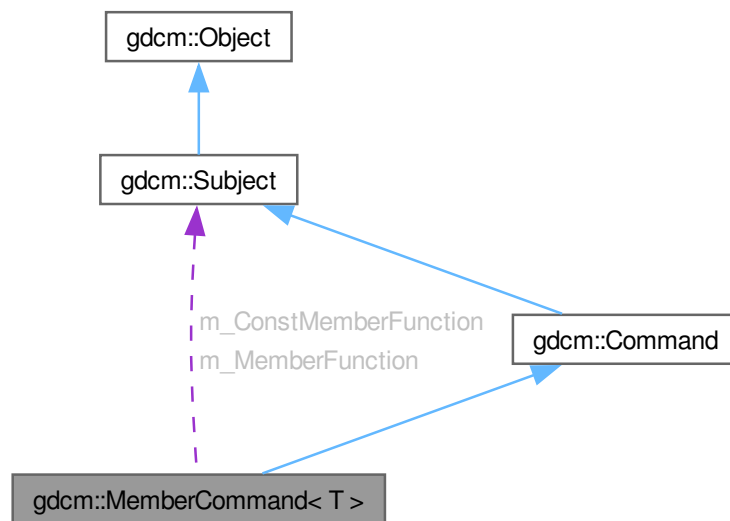
[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for `gdc::MemberCommand< T >`:



Collaboration diagram for `gdcmm::MemberCommand< T >`:



Public Types

- typedef [MemberCommand](#) Self
- typedef void(T::* [TConstMemberFunctionPointer](#)) (const [Subject](#) *, const [Event](#) &)
- typedef void(T::* [TMemberFunctionPointer](#)) ([Subject](#) *, const [Event](#) &)

Public Member Functions

- `MemberCommand` (const `Self` &)=delete
- void `Execute` (const `Subject` *caller, const `Event` &event) override
- void `Execute` (`Subject` *caller, const `Event` &event) override
- void `operator=` (const `Self` &)=delete
- void `SetCallbackFunction` (T *object, `TConstMemberFunctionPointer` memberFunction)
- void `SetCallbackFunction` (T *object, `TMemberFunctionPointer` memberFunction)

Public Member Functions inherited from `gdcm::Command`

- `Command` (const `Command` &)=delete
- void `operator=` (const `Command` &)=delete

Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [MemberCommand](#) > [New](#) ()

Protected Member Functions

- [MemberCommand](#) ()
- [~MemberCommand](#) () override=default

Protected Member Functions inherited from [gdcm::Command](#)

- [Command](#) ()
- [~Command](#) () override

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes

- [TConstMemberFunctionPointer](#) [m_ConstMemberFunction](#)
- [TMemberFunctionPointer](#) [m_MemberFunction](#)
- T * [m_This](#)

12.190.1 Detailed Description

```
template<class T>
class gdcM::MemberCommand< T >
```

[Command](#) subclass that calls a pointer to a member function.

[MemberCommand](#) calls a pointer to a member function with the same arguments as `Execute` on [Command](#).

12.190.2 Member Typedef Documentation

12.190.2.1 Self

```
template<class T>
typedef MemberCommand gdcM::MemberCommand< T >::Self
```

Standard class typedefs.

12.190.2.2 TConstMemberFunctionPointer

```
template<class T>
typedef void(T::* gdcM::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const Event &)
```

12.190.2.3 TMemberFunctionPointer

```
template<class T>
typedef void(T::* gdcM::MemberCommand< T >::TMemberFunctionPointer) (Subject *, const Event &)
```

pointer to a member function that takes a `Subject*` and the event

12.190.3 Constructor & Destructor Documentation

12.190.3.1 MemberCommand() [1/2]

```
template<class T>
gdcM::MemberCommand< T >::MemberCommand (
    const Self & ) [delete]
```

12.190.3.2 MemberCommand() [2/2]

```
template<class T>
gdcM::MemberCommand< T >::MemberCommand () [inline], [protected]
```

12.190.3.3 ~MemberCommand()

```
template<class T>
gdcM::MemberCommand< T >::~~MemberCommand ()  [override], [protected], [default]
```

12.190.4 Member Function Documentation

12.190.4.1 Execute() [1/2]

```
template<class T>
void gdcM::MemberCommand< T >::Execute (
    const Subject * caller,
    const Event & event)  [inline], [override], [virtual]
```

Invoke the member function with a const object.

Implements [gdcM::Command](#).

12.190.4.2 Execute() [2/2]

```
template<class T>
void gdcM::MemberCommand< T >::Execute (
    Subject * caller,
    const Event & event)  [inline], [override], [virtual]
```

Invoke the member function.

Implements [gdcM::Command](#).

12.190.4.3 New()

```
template<class T>
SmartPointer< MemberCommand > gdcM::MemberCommand< T >::New ()  [inline], [static]
```

Method for creation through the object factory.

12.190.4.4 operator=()

```
template<class T>
void gdcM::MemberCommand< T >::operator= (
    const Self & )  [delete]
```


12.190.4.5 SetCallbackFunction() [1/2]

```
template<class T>
void gdcM::MemberCommand< T >::SetCallbackFunction (
    T * object,
    TConstMemberFunctionPointer memberFunction) [inline]
```

12.190.4.6 SetCallbackFunction() [2/2]

```
template<class T>
void gdcM::MemberCommand< T >::SetCallbackFunction (
    T * object,
    TMemberFunctionPointer memberFunction) [inline]
```

Run-time type information (and related methods). Set the callback function along with the object that it will be invoked on.

12.190.5 Member Data Documentation

12.190.5.1 m_ConstMemberFunction

```
template<class T>
TConstMemberFunctionPointer gdcM::MemberCommand< T >::m_ConstMemberFunction [protected]
```

12.190.5.2 m_MemberFunction

```
template<class T>
TMemberFunctionPointer gdcM::MemberCommand< T >::m_MemberFunction [protected]
```

12.190.5.3 m_This

```
template<class T>
T* gdcM::MemberCommand< T >::m_This [protected]
```

The documentation for this class was generated from the following file:

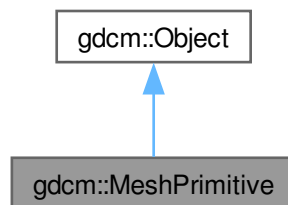
- [gdcMCommand.h](#)

12.191 gdcmmeshprimitive Class Reference

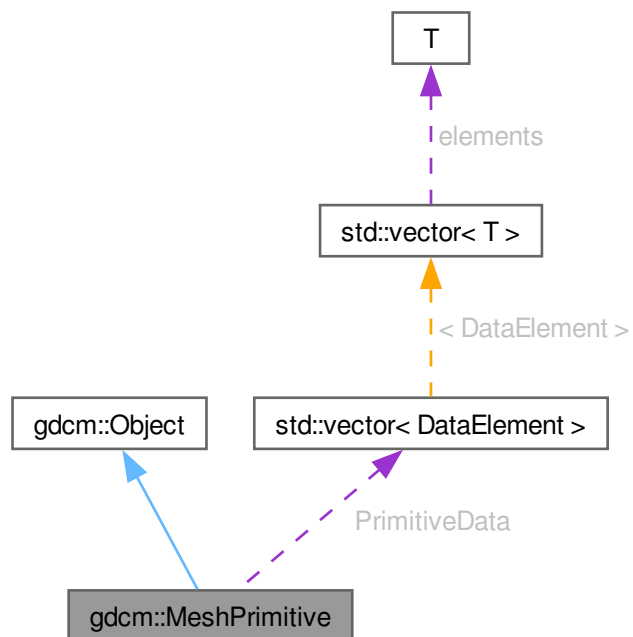
This class defines surface mesh primitives.

```
#include <gdcmmeshprimitive.h>
```

Inheritance diagram for gdcmmeshprimitive:



Collaboration diagram for gdcmmeshprimitive:



Public Types

- enum [MPType](#) {
[VERTEX](#) = 0 ,
[EDGE](#) ,
[TRIANGLE](#) ,
[TRIANGLE_STRIP](#) ,
[TRIANGLE_FAN](#) ,
[LINE](#) ,
[FACET](#) ,
[MPType_END](#) }

This enumeration defines primitive types.

- typedef std::vector< [DataElement](#) > [PrimitivesData](#)

Public Member Functions

- [MeshPrimitive](#) ()
- [~MeshPrimitive](#) () override
- void [AddPrimitiveData](#) ([DataElement](#) const &de)
- unsigned int [GetNumberOfPrimitivesData](#) () const
- [DataElement](#) & [GetPrimitiveData](#) ()
- const [DataElement](#) & [GetPrimitiveData](#) () const
- [DataElement](#) & [GetPrimitiveData](#) (const unsigned int idx)
- const [DataElement](#) & [GetPrimitiveData](#) (const unsigned int idx) const
- [PrimitivesData](#) & [GetPrimitivesData](#) ()
- const [PrimitivesData](#) & [GetPrimitivesData](#) () const
- [MPType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPType](#) type)

Public Member Functions inherited from [gdcM::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [MPType](#) [GetMPType](#) (const char *type)
- static const char * [GetMPTypeString](#) (const [MPType](#) type)

Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPType](#) [PrimitiveType](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.191.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

12.191.2 Member Typedef Documentation

12.191.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

12.191.3 Member Enumeration Documentation

12.191.3.1 MPType

```
enum gdcm::MeshPrimitive::MPType
```

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

VERTEX	
EDGE	
TRIANGLE	
TRIANGLE_STRIP	
TRIANGLE_FAN	
LINE	
FACET	
MPTType_END	

12.191.4 Constructor & Destructor Documentation

12.191.4.1 MeshPrimitive()

```
gdcM::MeshPrimitive::MeshPrimitive ()
```

12.191.4.2 ~MeshPrimitive()

```
gdcM::MeshPrimitive::~~MeshPrimitive () [override]
```

12.191.5 Member Function Documentation

12.191.5.1 AddPrimitiveData()

```
void gdcM::MeshPrimitive::AddPrimitiveData (
    DataElement const & de)
```

12.191.5.2 GetMPTType()

```
MPTType gdcM::MeshPrimitive::GetMPTType (
    const char * type) [static]
```

12.191.5.3 GetMPTTypeString()

```
const char * gdcM::MeshPrimitive::GetMPTTypeString (
    const MPTType type) [static]
```

12.191.5.4 GetNumberOfPrimitivesData()

unsigned int gdcmmeshprimitive::GetNumberOfPrimitivesData () const

12.191.5.5 GetPrimitiveData() [1/4]

[DataElement](#) & gdcmmeshprimitive::GetPrimitiveData ()

12.191.5.6 GetPrimitiveData() [2/4]

const [DataElement](#) & gdcmmeshprimitive::GetPrimitiveData () const

12.191.5.7 GetPrimitiveData() [3/4]

[DataElement](#) & gdcmmeshprimitive::GetPrimitiveData (
const unsigned int idx)

12.191.5.8 GetPrimitiveData() [4/4]

const [DataElement](#) & gdcmmeshprimitive::GetPrimitiveData (
const unsigned int idx) const

12.191.5.9 GetPrimitivesData() [1/2]

[PrimitivesData](#) & gdcmmeshprimitive::GetPrimitivesData ()

12.191.5.10 GetPrimitivesData() [2/2]

const [PrimitivesData](#) & gdcmmeshprimitive::GetPrimitivesData () const

12.191.5.11 GetPrimitiveType()

[MPTType](#) gdcmmeshprimitive::GetPrimitiveType () const

12.191.5.12 SetPrimitiveData() [1/2]

void gdcmmeshprimitive::SetPrimitiveData (
const unsigned int idx,
[DataElement](#) const & de)

12.191.5.13 SetPrimitiveData() [2/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (  
    DataElement const & de)
```

12.191.5.14 SetPrimitivesData()

```
void gdcm::MeshPrimitive::SetPrimitivesData (  
    PrimitivesData const & DEs)
```

12.191.5.15 SetPrimitiveType()

```
void gdcm::MeshPrimitive::SetPrimitiveType (  
    const MPType type)
```

12.191.6 Member Data Documentation

12.191.6.1 PrimitiveData

[PrimitivesData](#) gdcm::MeshPrimitive::PrimitiveData [protected]

12.191.6.2 PrimitiveType

[MPType](#) gdcm::MeshPrimitive::PrimitiveType [protected]

The documentation for this class was generated from the following file:

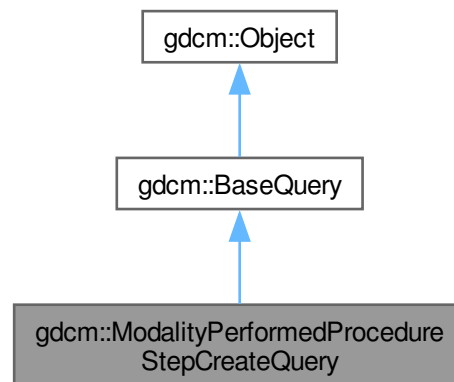
- [gdcmMeshPrimitive.h](#)

12.192 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

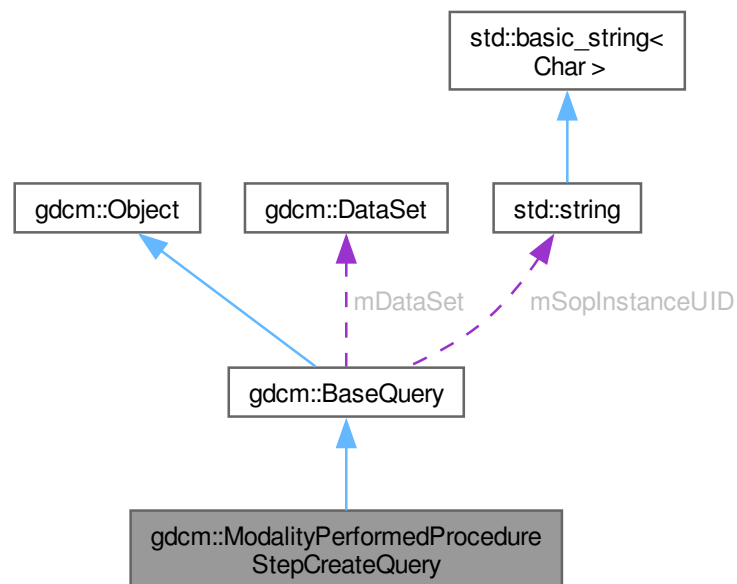
[ModalityPerformedProcedureStepCreateQuery](#).

```
#include <gdcmModalityPerformedProcedureStepCreateQuery.h>
```

Inheritance diagram for `gdcm::ModalityPerformedProcedureStepCreateQuery`:



Collaboration diagram for `gdcm::ModalityPerformedProcedureStepCreateQuery`:



Public Member Functions

- [ModalityPerformedProcedureStepCreateQuery](#) (const `std::string` &`iSopInstanceUID`)

- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet mDataSet](#)
- `std::string mSopInstanceUID`

12.192.1 Detailed Description

[ModalityPerformedProcedureStepCreateQuery](#).

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

12.192.2 Constructor & Destructor Documentation

12.192.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
    const std::string & iSopInstanceUID)
```

12.192.3 Member Function Documentation

12.192.3.1 GetAbstractSyntaxUID()

[UIDs::TSName](#) `gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID () const` [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.192.3.2 GetRequiredDataSet()

[gdcm::DataSet](#) `gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet () const`

12.192.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
    bool inStrict = true) const
```

 [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.192.4 Friends And Related Symbol Documentation

12.192.4.1 QueryFactory

friend class QueryFactory [friend]

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

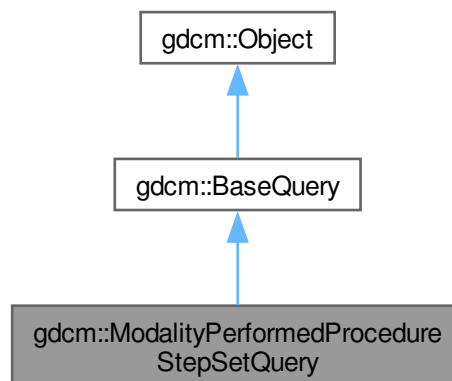
- [gdcmModalityPerformedProcedureStepCreateQuery.h](#)

12.193 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

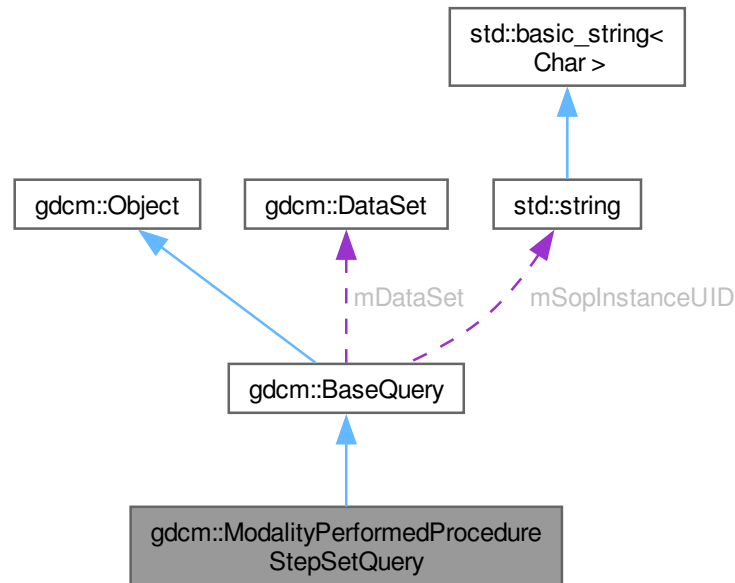
[ModalityPerformedProcedureStepSetQuery](#).

```
#include <gdcmModalityPerformedProcedureStepSetQuery.h>
```

Inheritance diagram for gdcm::ModalityPerformedProcedureStepSetQuery:



Collaboration diagram for `gdcm::ModalityPerformedProcedureStepSetQuery`:



Public Member Functions

- [ModalityPerformedProcedureStepSetQuery](#) (const std::string &iSopInstanceUID)
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
 - void [AddQueryDataSet](#) (const [DataSet](#) &ds)
 - [DataSet](#) & [GetQueryDataSet](#) ()
 - [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
 - void [Print](#) (std::ostream &os) const override
 - void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
 - void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
 - void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
 - const std::ostream & [WriteHelpFile](#) (std::ostream &os)
 - bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.193.1 Detailed Description

[ModalityPerformedProcedureStepSetQuery](#).

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

12.193.2 Constructor & Destructor Documentation

12.193.2.1 [ModalityPerformedProcedureStepSetQuery](#)()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
    const std::string & iSopInstanceUID)
```

12.193.3 Member Function Documentation

12.193.3.1 GetAbstractSyntaxUID()

[UIDs::TSName](#) `gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID () const` [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.193.3.2 GetRequiredDataSet()

[gdcm::DataSet](#) `gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet () const`

12.193.3.3 ValidateQuery()

`bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (`
 `bool inStrict = true) const` [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.193.4 Friends And Related Symbol Documentation

12.193.4.1 QueryFactory

`friend class QueryFactory` [friend]

References [QueryFactory](#).

Referenced by [QueryFactory](#).

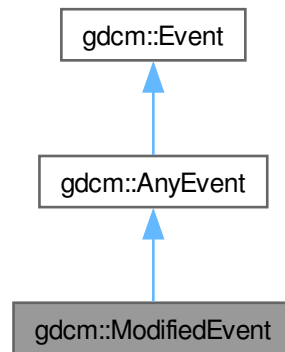
The documentation for this class was generated from the following file:

- [gdcmModalityPerformedProcedureStepSetQuery.h](#)

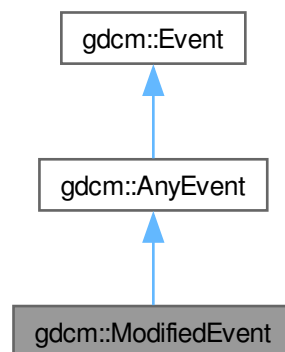
12.194 gdcm::ModifiedEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()

- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmlEvent.h](#)

12.195 gdcml::Module Class Reference

Class for representing a [Module](#).

```
#include <gdcmlModule.h>
```

Public Types

- typedef std::vector< std::string > [ArrayIncludeMacrosType](#)
- typedef std::map< [Tag](#), [ModuleEntry](#) > [MapModuleEntry](#)

Public Member Functions

- [Module](#) ()=default
- void [AddMacro](#) (const char *include)
- void [AddModuleEntry](#) (const [Tag](#) &tag, const [ModuleEntry](#) &module)
Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.
- void [Clear](#) ()
- bool [FindModuleEntryInMacros](#) ([Macros](#) const ¯os, const [Tag](#) &tag) const
- const [ModuleEntry](#) & [GetModuleEntryInMacros](#) ([Macros](#) const ¯os, const [Tag](#) &tag) const
- const char * [GetName](#) () const
- void [SetName](#) (const char *name)
- bool [Verify](#) (const [DataSet](#) &ds, [Usage](#) const &usage) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Module](#) &_val)

12.195.1 Detailed Description

Class for representing a [Module](#).

Note

[Module](#): A set of Attributes within an Information Entity or Normalized [IOD](#) which are logically related to each other.

See also

[Macro](#)

Examples

[TraverseModules.cxx](#).

12.195.2 Member Typedef Documentation

12.195.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcmm::Module::ArrayIncludeMacrosType
```

12.195.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcmm::Module::MapModuleEntry
```

12.195.3 Constructor & Destructor Documentation

12.195.3.1 Module()

```
gdcmm::Module::Module () [default]
```

References [Module\(\)](#), and [operator<<](#).

Referenced by [Module\(\)](#), and [operator<<](#).

12.195.4 Member Function Documentation

12.195.4.1 AddMacro()

```
void gdcmm::Module::AddMacro (  
    const char * include) [inline]
```

12.195.4.2 AddModuleEntry()

```
void gdcmm::Module::AddModuleEntry (
    const Tag & tag,
    const ModuleEntry & module) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

12.195.4.3 Clear()

```
void gdcmm::Module::Clear () [inline]
```

12.195.4.4 FindModuleEntryInMacros()

```
bool gdcmm::Module::FindModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag) const
```

Find or Get a [ModuleEntry](#). [ModuleEntry](#) are either search are root-level or within nested-macro included in module.

Examples

[TraverseModules.cxx](#).

12.195.4.5 GetModuleEntryInMacros()

```
const ModuleEntry & gdcmm::Module::GetModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag) const
```

12.195.4.6 GetName()

```
const char * gdcmm::Module::GetName () const [inline]
```

12.195.4.7 SetName()

```
void gdcmm::Module::SetName (
    const char * name) [inline]
```

12.195.4.8 Verify()

```
bool gdcmm::Module::Verify (
    const DataSet & ds,
    Usage const & usage) const
```

12.195.5 Friends And Related Symbol Documentation

12.195.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Module & _val) [friend]
```

References [Module\(\)](#).

Referenced by [Module\(\)](#).

The documentation for this class was generated from the following file:

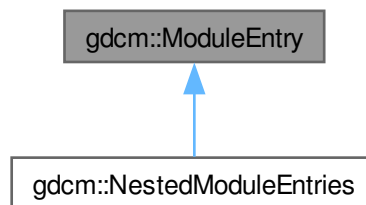
- [gdcmModule.h](#)

12.196 gdcm::ModuleEntry Class Reference

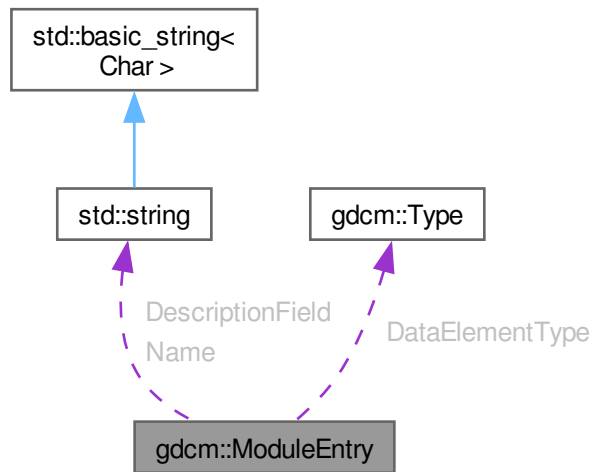
Class for representing a [ModuleEntry](#).

```
#include <gdcmModuleEntry.h>
```

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for `gdcm::ModuleEntry`:



Public Types

- typedef `std::string` [Description](#)

Public Member Functions

- [ModuleEntry](#) (`const char *name=""`, `const char *type="3"`, `const char *description=""`)
- virtual `~ModuleEntry` ()=default
- const [Description](#) & [GetDescription](#) () const
- const char * [GetName](#) () const
- const [Type](#) & [GetType](#) () const
- void [SetDescription](#) (`const char *d`)
- void [SetName](#) (`const char *name`)
- void [SetType](#) (`const Type &type`)

Protected Attributes

- [Type](#) [DataElementType](#)
- [Description](#) [DescriptionField](#)
- `std::string` [Name](#)

Friends

- `std::ostream & operator<<` (`std::ostream &_os`, const [ModuleEntry](#) &_val)

12.196.1 Detailed Description

Class for representing a [ModuleEntry](#).

Note

bla

See also

[DictEntry](#)

Examples

[TraverseModules.cxx](#).

12.196.2 Member Typedef Documentation

12.196.2.1 Description

typedef std::string [gdcmmoduleentry::Description](#)

12.196.3 Constructor & Destructor Documentation

12.196.3.1 ModuleEntry()

```
gdcmmoduleentry::ModuleEntry (
    const char * name = "",
    const char * type = "3",
    const char * description = "") [inline]
```

References [DataElementType](#), [DescriptionField](#), [gdcmmoduleentry::Type::GetTypeType\(\)](#), and [Name](#).

Referenced by [gdcmmoduleentry::NestedModuleEntries::NestedModuleEntries\(\)](#), [~ModuleEntry\(\)](#), [gdcmmoduleentry::NestedModuleEntries::AddModuleEntry\(\)](#), [gdcmmoduleentry::NestedModuleEntries::GetModuleEntry\(\)](#), [gdcmmoduleentry::NestedModuleEntries::GetModuleEntry\(\)](#), and [operator<<](#).

12.196.3.2 ~ModuleEntry()

```
virtual gdcmmoduleentry::~ModuleEntry () [virtual], [default]
```

References [ModuleEntry\(\)](#), and [operator<<](#).

12.196.4 Member Function Documentation

12.196.4.1 GetDescription()

const [Description](#) & gdcm::ModuleEntry::GetDescription () const [inline]

References [DescriptionField](#).

12.196.4.2 GetName()

const char * gdcm::ModuleEntry::GetName () const [inline]

References [Name](#).

12.196.4.3 GetType()

const [Type](#) & gdcm::ModuleEntry::GetType () const [inline]

Examples

[TraverseModules.cxx](#).

References [DataElementType](#).

12.196.4.4 SetDescription()

void gdcm::ModuleEntry::SetDescription (
 const char * d) [inline]

References [DescriptionField](#).

12.196.4.5 SetName()

void gdcm::ModuleEntry::SetName (
 const char * name) [inline]

References [Name](#).

12.196.4.6 SetType()

void gdcm::ModuleEntry::SetType (
 const [Type](#) & type) [inline]

References [DataElementType](#).

12.196.5 Friends And Related Symbol Documentation

12.196.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const ModuleEntry & __val) [friend]
```

References [ModuleEntry\(\)](#), [DataElementType](#), [DescriptionField](#), and [Name](#).

Referenced by [~ModuleEntry\(\)](#).

12.196.6 Member Data Documentation

12.196.6.1 DataElementType

[Type](#) gdcm::ModuleEntry::DataElementType [protected]

Referenced by [ModuleEntry\(\)](#), [GetType\(\)](#), [operator<<](#), [gdcm::NestedModuleEntries::operator<<](#), and [SetType\(\)](#).

12.196.6.2 DescriptionField

[Description](#) gdcm::ModuleEntry::DescriptionField [protected]

Referenced by [ModuleEntry\(\)](#), [GetDescription\(\)](#), [operator<<](#), [gdcm::NestedModuleEntries::operator<<](#), and [SetDescription\(\)](#).

12.196.6.3 Name

std::string gdcm::ModuleEntry::Name [protected]

Referenced by [ModuleEntry\(\)](#), [GetName\(\)](#), [operator<<](#), [gdcm::NestedModuleEntries::operator<<](#), and [SetName\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmModuleEntry.h](#)

12.197 gdcm::Modules Class Reference

Class for representing a [Modules](#).

```
#include <gdcmModules.h>
```

Public Types

- typedef std::map< std::string, [Module](#) > [ModuleMapType](#)

Public Member Functions

- [Modules](#) ()=default
- void [AddModule](#) (const char *ref, const [Module](#) &module)
- void [Clear](#) ()
- const [Module](#) & [GetModule](#) (const char *name) const
- bool [IsEmpty](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Modules](#) &_val)

12.197.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples

[TraverseModules.cxx](#).

12.197.2 Member Typedef Documentation

12.197.2.1 ModuleMapType

typedef std::map<std::string, [Module](#)> [gdcmm::Modules::ModuleMapType](#)

12.197.3 Constructor & Destructor Documentation

12.197.3.1 Modules()

[gdcmm::Modules::Modules](#) () [default]

References [Modules\(\)](#), and [operator<<](#).

Referenced by [Modules\(\)](#), and [operator<<](#).

12.197.4 Member Function Documentation

12.197.4.1 AddModule()

```
void gdcmm::Modules::AddModule (
    const char * ref,
    const Module & module) [inline]
```

References [gdcmm_assert](#).

12.197.4.2 Clear()

```
void gdcmm::Modules::Clear () [inline]
```

12.197.4.3 GetModule()

```
const Module & gdcmm::Modules::GetModule (
    const char * name) const [inline]
```

Examples

[TraverseModules.cxx](#).

References [gdcmm_assert](#).

12.197.4.4 IsEmpty()

```
bool gdcmm::Modules::IsEmpty () const [inline]
```

12.197.5 Friends And Related Symbol Documentation

12.197.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const Modules & __val) [friend]
```

References [Modules\(\)](#).

Referenced by [Modules\(\)](#).

The documentation for this class was generated from the following file:

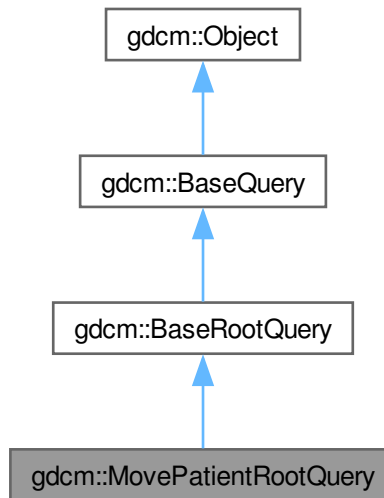
- [gdcmmModules.h](#)

12.198 gdcM::MovePatientRootQuery Class Reference

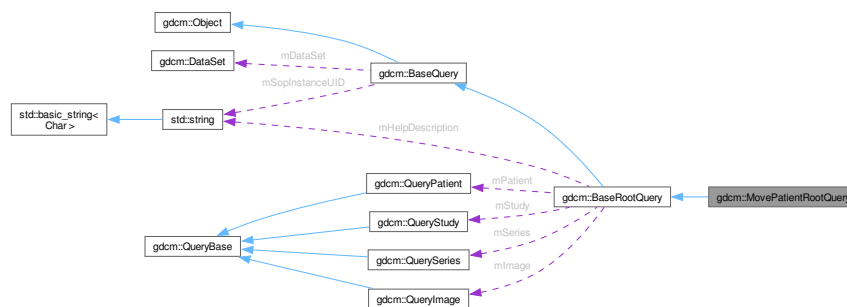
[MovePatientRootQuery](#).

```
#include <gdcMMovePatientRootQuery.h>
```

Inheritance diagram for gdcM::MovePatientRootQuery:



Collaboration diagram for gdcM::MovePatientRootQuery:



Public Member Functions

- [MovePatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- `std::vector< Tag >` [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) * [Construct](#) (ERootType inRootType, EQueryLevel qlvel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) (EQueryLevel ql)

Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.198.1 Detailed Description

[MovePatientRootQuery](#).

contains: the class which will produce a dataset for c-move with patient root

12.198.2 Constructor & Destructor Documentation

12.198.2.1 [MovePatientRootQuery](#)()

`gdcm::MovePatientRootQuery::MovePatientRootQuery ()`

12.198.3 Member Function Documentation

12.198.3.1 [GetAbstractSyntaxUID](#)()

`UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID () const` [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.198.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcmm::MovePatientRootQuery::GetTagListByLevel (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcmm::BaseRootQuery](#).

12.198.3.3 InitializeDataSet()

```
void gdcmm::MovePatientRootQuery::InitializeDataSet (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implements [gdcmm::BaseRootQuery](#).

12.198.3.4 ValidateQuery()

```
bool gdcmm::MovePatientRootQuery::ValidateQuery (  
    bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcmm::BaseRootQuery](#).

12.198.4 Friends And Related Symbol Documentation

12.198.4.1 QueryFactory

friend class QueryFactory [friend]

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

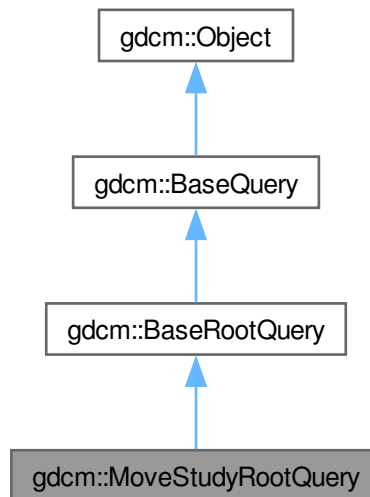
- [gdcmmMovePatientRootQuery.h](#)

12.199 gdcmm::MoveStudyRootQuery Class Reference

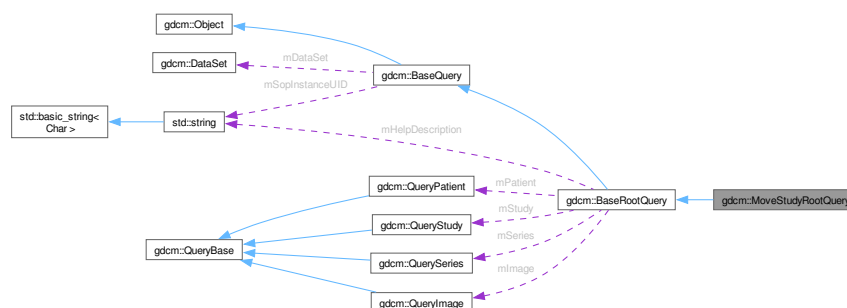
[MoveStudyRootQuery](#).

```
#include <gdcmmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcmm::MoveStudyRootQuery:



Collaboration diagram for gdcmm::MoveStudyRootQuery:



Public Member Functions

- [MoveStudyRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- `std::vector< Tag >` [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) (ERootType roottype)

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet](#) & [GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
Set/Get the internal representation of the query as a [DataSet](#).
- std::string [GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Friends

- class [QueryFactory](#)

Additional Inherited Members

Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) * [Construct](#) (ERootType inRootType, EQueryLevel qlvel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) (EQueryLevel ql)

Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.199.1 Detailed Description

[MoveStudyRootQuery](#).

contains: the class which will produce a dataset for C-MOVE with study root

12.199.2 Constructor & Destructor Documentation

12.199.2.1 [MoveStudyRootQuery](#)()

[gdcm::MoveStudyRootQuery::MoveStudyRootQuery](#) ()

12.199.3 Member Function Documentation

12.199.3.1 [GetAbstractSyntaxUID](#)()

[UIDs::TSName](#) [gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID](#) () const [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.199.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcmm::MoveStudyRootQuery::GetTagListByLevel (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcmm::BaseRootQuery](#).

12.199.3.3 InitializeDataSet()

```
void gdcmm::MoveStudyRootQuery::InitializeDataSet (  
    const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implements [gdcmm::BaseRootQuery](#).

12.199.3.4 ValidateQuery()

```
bool gdcmm::MoveStudyRootQuery::ValidateQuery (  
    bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcmm::BaseRootQuery](#).

12.199.4 Friends And Related Symbol Documentation

12.199.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

- [gdcmmMoveStudyRootQuery.h](#)

12.200 gdcm::MrProtocol Class Reference

Class for [MrProtocol](#).

```
#include <gdcmMrProtocol.h>
```

Classes

- struct [Slice](#)
- struct [SliceArray](#)
- struct [Vector3](#)

Public Member Functions

- [MrProtocol](#) ()
- [~MrProtocol](#) ()
- bool [FindMrProtocolByName](#) (const char *name) const
- const char * [GetMrProtocolByName](#) (const char *name) const
- bool [GetSliceArray](#) ([MrProtocol::SliceArray](#) &sa) const
- int [GetVersion](#) () const
- bool [Load](#) (const [ByteValue](#) *bv, const char *str, int version)
- void [Print](#) (std::ostream &os) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [MrProtocol](#) &d)

12.200.1 Detailed Description

Class for [MrProtocol](#).

Examples

[MrProtocol.cxx](#).

12.200.2 Constructor & Destructor Documentation

12.200.2.1 MrProtocol()

```
gdcm::MrProtocol::MrProtocol ()
```

Referenced by [operator<<](#).

12.200.2.2 ~MrProtocol()

```
gdcm::MrProtocol::~~MrProtocol ()
```

12.200.3 Member Function Documentation

12.200.3.1 FindMrProtocolByName()

```
bool gdcm::MrProtocol::FindMrProtocolByName (
    const char * name) const
```

12.200.3.2 GetMrProtocolByName()

```
const char * gdcm::MrProtocol::GetMrProtocolByName (
    const char * name) const
```

12.200.3.3 GetSliceArray()

```
bool gdcm::MrProtocol::GetSliceArray (
    MrProtocol::SliceArray & sa) const
```

12.200.3.4 GetVersion()

```
int gdcm::MrProtocol::GetVersion () const
```

12.200.3.5 Load()

```
bool gdcm::MrProtocol::Load (
    const ByteValue * bv,
    const char * str,
    int version)
```

12.200.3.6 Print()

```
void gdcm::MrProtocol::Print (
    std::ostream & os) const
```

Referenced by [operator<<](#).

12.200.4 Friends And Related Symbol Documentation

12.200.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const MrProtocol & d) [friend]
```

References [MrProtocol\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

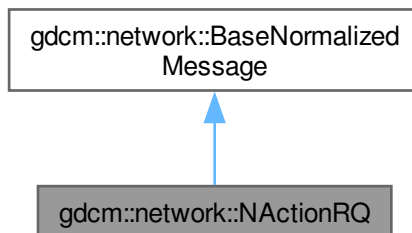
- [gdcmmrprotocol.h](#)

12.201 gdcmm::network::NActionRQ Class Reference

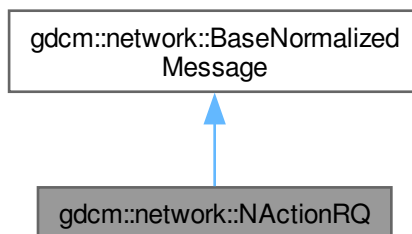
[NActionRQ](#).

```
#include <gdcmmActionMessages.h>
```

Inheritance diagram for gdcmm::network::NActionRQ:



Collaboration diagram for gdcmm::network::NActionRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`) override

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`

12.201.1 Detailed Description

[NActionRQ](#).

this file defines the messages for the NAction action

12.201.2 Member Function Documentation

12.201.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NActionRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

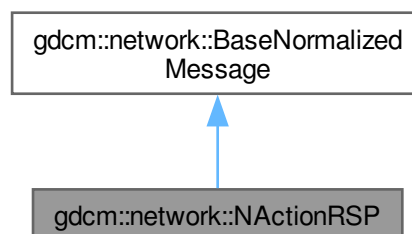
- [gdcmNActionMessages.h](#)

12.202 gdcm::network::NActionRSP Class Reference

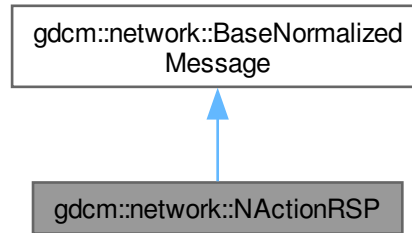
[NActionRSP](#) this file defines the messages for the NAction action.

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for `gdcm::network::NActionRSP`:



Collaboration diagram for `gdcm::network::NActionRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)=0

12.202.1 Detailed Description

[NActionRSP](#) this file defines the messages for the NAction action.

12.202.2 Member Function Documentation

12.202.2.1 `ConstructPDVByDataSet()`

```
std::vector< PresentationDataValue > gdcm::network::NActionRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

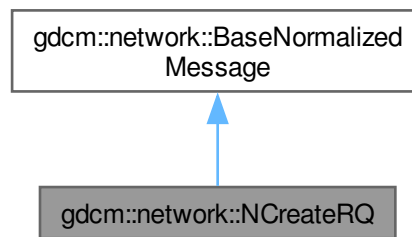
- [gdcmNActionMessages.h](#)

12.203 gdcn::network::NCreateRQ Class Reference

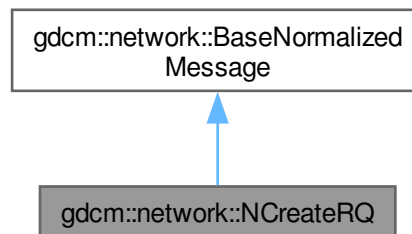
[NCreateRQ](#).

```
#include <gdcnNCreateMessages.h>
```

Inheritance diagram for gdcn::network::NCreateRQ:



Collaboration diagram for gdcn::network::NCreateRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery) override

Public Member Functions inherited from [gdcn::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage ()`=default

12.203.1 Detailed Description

[NCreateRQ](#).

this file defines the messages for the ncreate action

12.203.2 Member Function Documentation

12.203.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::NCreateRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcmm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

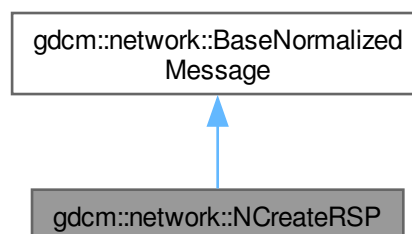
- [gdcmmNCreateMessages.h](#)

12.204 gdcmm::network::NCreateRSP Class Reference

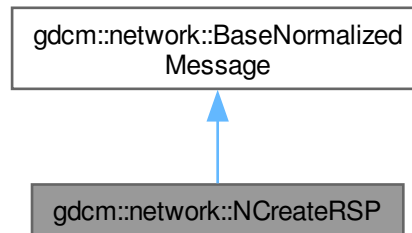
[NCreateRSP](#) this file defines the messages for the ncreate action.

```
#include <gdcmmNCreateMessages.h>
```

Inheritance diagram for gdcmm::network::NCreateRSP:



Collaboration diagram for gdcm::network::NCreateRSP:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)=0

12.204.1 Detailed Description

[NCreateRSP](#) this file defines the messages for the ncreate action.

12.204.2 Member Function Documentation

12.204.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

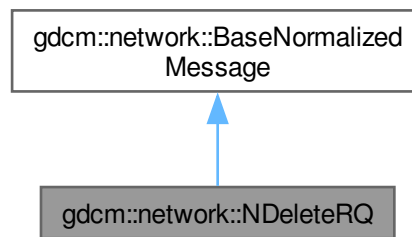
- [gdcmNCreateMessages.h](#)

12.205 gdcmm::network::NDeleteRQ Class Reference

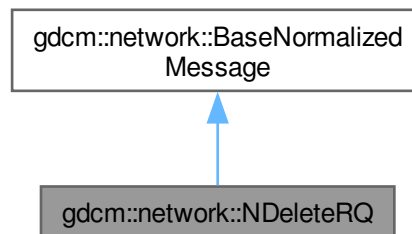
[NDeleteRQ](#).

```
#include <gdcmmNDeleteMessages.h>
```

Inheritance diagram for gdcmm::network::NDeleteRQ:



Collaboration diagram for gdcmm::network::NDeleteRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery) override

Public Member Functions inherited from [gdcmm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default

12.205.1 Detailed Description

[NDeleteRQ](#).

this file defines the messages for the ndelete action

12.205.2 Member Function Documentation

12.205.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

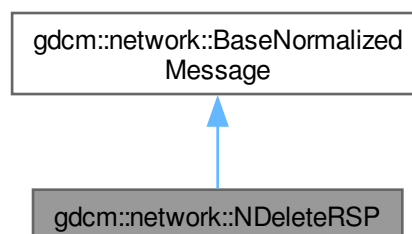
- [gdcmNDeleteMessages.h](#)

12.206 gdcm::network::NDeleteRSP Class Reference

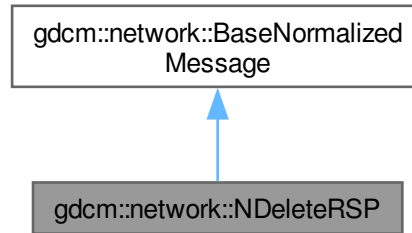
[NDeleteRSP](#) this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for gdcm::network::NDeleteRSP:



Collaboration diagram for `gdcm::network::NDeleteRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage` ()=default
- `virtual std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)=0

12.206.1 Detailed Description

[NDeleteRSP](#) this file defines the messages for the ndelete action.

12.206.2 Member Function Documentation

12.206.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

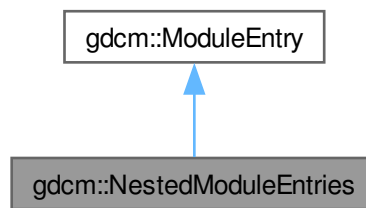
- [gdcmNDeleteMessages.h](#)

12.207 gdcm::NestedModuleEntries Class Reference

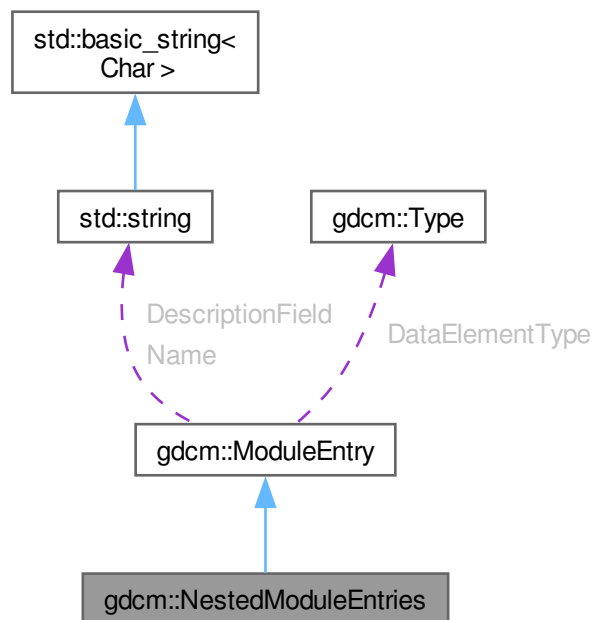
Class for representing a [NestedModuleEntries](#).

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

- typedef std::vector< [ModuleEntry](#) >::size_type [SizeType](#)

Public Types inherited from [gdcmm::ModuleEntry](#)

- typedef std::string [Description](#)

Public Member Functions

- [NestedModuleEntries](#) (const char *name="", const char *type="3", const char *description="")
- void [AddModuleEntry](#) (const [ModuleEntry](#) &me)
- [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx)
- const [ModuleEntry](#) & [GetModuleEntry](#) ([SizeType](#) idx) const
- [SizeType](#) [GetNumberOfModuleEntries](#) ()

Public Member Functions inherited from [gdcmm::ModuleEntry](#)

- [ModuleEntry](#) (const char *name="", const char *type="3", const char *description="")
- virtual [~ModuleEntry](#) ()=default
- const [Description](#) & [GetDescription](#) () const
- const char * [GetName](#) () const
- const [Type](#) & [GetType](#) () const
- void [SetDescription](#) (const char *d)
- void [SetName](#) (const char *name)
- void [SetType](#) (const [Type](#) &type)

Friends

- std::ostream & [operator<<](#) (std::ostream &__os, const [NestedModuleEntries](#) &__val)

Additional Inherited Members

Protected Attributes inherited from [gdcmm::ModuleEntry](#)

- [Type](#) [DataElementType](#)
- [Description](#) [DescriptionField](#)
- std::string [Name](#)

12.207.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

Note

bla

See also

[ModuleEntry](#)

12.207.2 Member Typedef Documentation

12.207.2.1 SizeType

typedef std::vector<[ModuleEntry](#)>::size_type [gdcm::NestedModuleEntries::SizeType](#)

12.207.3 Constructor & Destructor Documentation

12.207.3.1 NestedModuleEntries()

```
gdcm::NestedModuleEntries::NestedModuleEntries (  
    const char * name = "",  
    const char * type = "3",  
    const char * description = "") [inline]
```

References [gdcm::ModuleEntry::ModuleEntry\(\)](#).

Referenced by [operator<<](#).

12.207.4 Member Function Documentation

12.207.4.1 AddModuleEntry()

```
void gdcm::NestedModuleEntries::AddModuleEntry (  
    const ModuleEntry & me) [inline]
```

References [gdcm::ModuleEntry::ModuleEntry\(\)](#).

12.207.4.2 GetModuleEntry() [1/2]

```
ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (  
    SizeType idx) [inline]
```

References [gdcm::ModuleEntry::ModuleEntry\(\)](#).

12.207.4.3 GetModuleEntry() [2/2]

```
const ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (  
    SizeType idx) const [inline]
```

References [gdcm::ModuleEntry::ModuleEntry\(\)](#).

12.207.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries () [inline]
```

12.207.5 Friends And Related Symbol Documentation

12.207.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const NestedModuleEntries & __val) [friend]
```

References [NestedModuleEntries\(\)](#), [gdcm::ModuleEntry::DataElementType](#), [gdcm::ModuleEntry::DescriptionField](#), and [gdcm::ModuleEntry::Name](#).

The documentation for this class was generated from the following file:

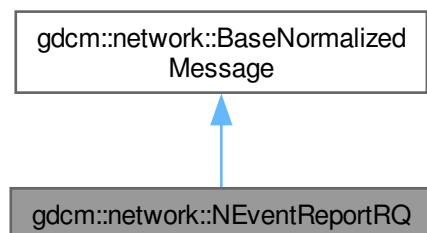
- [gdcmNestedModuleEntries.h](#)

12.208 gdcm::network::NEventReportRQ Class Reference

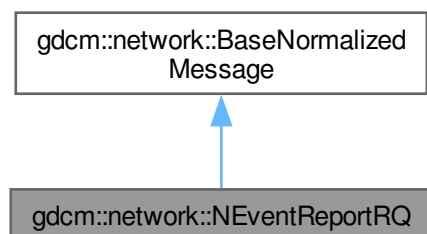
[NEventReportRQ](#).

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRQ:



Collaboration diagram for gdcm::network::NEventReportRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`) override

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`

12.208.1 Detailed Description

[NEventReportRQ](#).

this file defines the messages for the neventreport action

12.208.2 Member Function Documentation

12.208.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

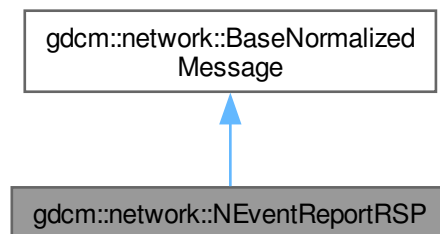
- [gdcmNEventReportMessages.h](#)

12.209 gdcm::network::NEventReportRSP Class Reference

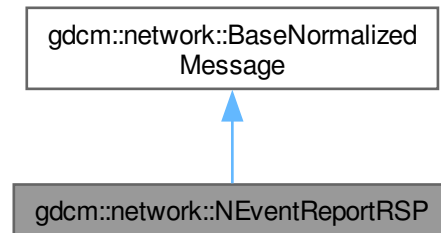
[NEventReportRSP](#) this file defines the messages for the neventreport action.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for `gdcm::network::NEventReportRSP`:



Collaboration diagram for `gdcm::network::NEventReportRSP`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (`const DataSet *inDataSet`)

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage` ()=default
- virtual `std::vector< PresentationDataValue > ConstructPDV` (`const ULConnection &inConnection`, `const BaseQuery *inQuery`)=0

12.209.1 Detailed Description

[NEventReportRSP](#) this file defines the messages for the neventreport action.

12.209.2 Member Function Documentation

12.209.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

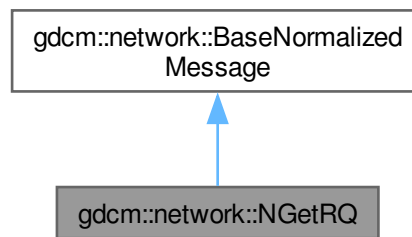
- [gdcmNEventReportMessages.h](#)

12.210 gdcm::network::NGetRQ Class Reference

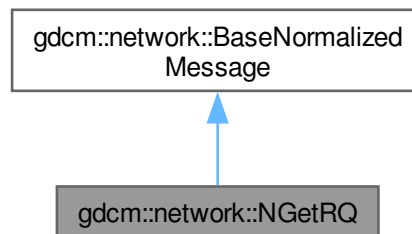
[NGetRQ](#).

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRQ:



Collaboration diagram for gdcm::network::NGetRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery) override

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage ()`=default

12.210.1 Detailed Description

[NGetRQ](#).

this file defines the messages for the nget action

12.210.2 Member Function Documentation

12.210.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::NGetRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcmm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

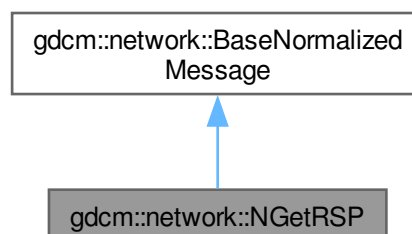
- [gdcmmNGetMessages.h](#)

12.211 gdcmm::network::NGetRSP Class Reference

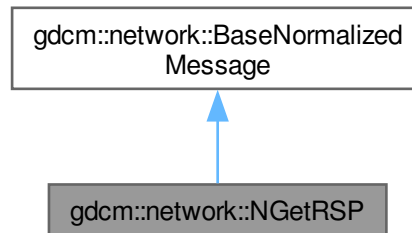
[NGetRSP](#) this file defines the messages for the nget action.

```
#include <gdcmmNGetMessages.h>
```

Inheritance diagram for gdcmm::network::NGetRSP:



Collaboration diagram for gdcm::network::NGetRSP:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

12.211.1 Detailed Description

[NGetRSP](#) this file defines the messages for the nget action.

12.211.2 Member Function Documentation

12.211.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NGetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

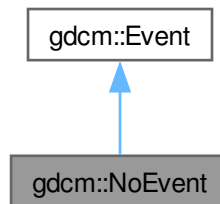
The documentation for this class was generated from the following file:

- [gdcmNGetMessages.h](#)

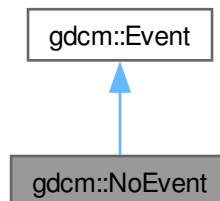
12.212 gdcmm::NoEvent Class Reference

```
#include <gdcmmEvent.h>
```

Inheritance diagram for gdcmm::NoEvent:



Collaboration diagram for gdcmm::NoEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcmm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

12.212.1 Detailed Description

Define some common GDCM events

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.213 gdcm::network::NormalizedMessageFactory Class Reference

```
#include <gdcmNormalizedMessageFactory.h>
```

Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructNAction](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNCreate](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNDelete](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNEventReport](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNGet](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructNSet](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)

12.213.1 Member Function Documentation

12.213.1.1 ConstructNAction()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNAction (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.213.1.2 ConstructNCreate()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNCreate (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.213.1.3 ConstructNDelete()

```
std::vector< PresentationDataValue > gdcmm::network::NormalizedMessageFactory::ConstructNDelete (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.213.1.4 ConstructNEventReport()

```
std::vector< PresentationDataValue > gdcmm::network::NormalizedMessageFactory::ConstructNEventReport (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.213.1.5 ConstructNGet()

```
std::vector< PresentationDataValue > gdcmm::network::NormalizedMessageFactory::ConstructNGet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.213.1.6 ConstructNSet()

```
std::vector< PresentationDataValue > gdcmm::network::NormalizedMessageFactory::ConstructNSet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

The documentation for this class was generated from the following file:

- [gdcmmNormalizedMessageFactory.h](#)

12.214 gdcmm::NormalizedNetworkFunctions Class Reference

Normalized Network Functions.

```
#include <gdcmmNormalizedNetworkFunctions.h>
```

Static Public Member Functions

- static [BaseQuery](#) * [ConstructQuery](#) (const std::string &sopInstanceUID, const [DataSet](#) &queryds, [ENQueryType](#) queryType=eCreateMMPS)
- static bool [NAction](#) (const char *remote, uint16_t portno, const [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)
- static bool [NCreate](#) (const char *remote, uint16_t portno, [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)
- static bool [NDelete](#) (const char *remote, uint16_t portno, const [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)
- static bool [NEventReport](#) (const char *remote, uint16_t portno, const [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)
- static bool [NGet](#) (const char *remote, uint16_t portno, const [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)
- static bool [NSet](#) (const char *remote, uint16_t portno, const [BaseQuery](#) *query, std::vector< [DataSet](#) > &retDataSets, const char *aetitle, const char *call)

12.214.1 Detailed Description

Normalized Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- N-EVENT-REPORT
- N-GET
- N-SET
- N-ACTION
- N-CREATE
- N-DELETE

12.214.2 Member Function Documentation

12.214.2.1 ConstructQuery()

```
BaseQuery * gdcm::NormalizedNetworkFunctions::ConstructQuery (  
    const std::string & sopInstanceUID,  
    const DataSet & queryds,  
    ENQueryType queryType = eCreateMMPS) [static]
```

References [gdcm::eCreateMMPS](#).

12.214.2.2 NAction()

```
bool gdcm::NormalizedNetworkFunctions::NAction (  
    const char * remote,  
    uint16_t portno,  
    const BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

12.214.2.3 NCreate()

```
bool gdcmm::NormalizedNetworkFunctions::NCreate (  
    const char * remote,  
    uint16_t portno,  
    BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

12.214.2.4 NDelete()

```
bool gdcmm::NormalizedNetworkFunctions::NDelete (  
    const char * remote,  
    uint16_t portno,  
    const BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

12.214.2.5 NEventReport()

```
bool gdcmm::NormalizedNetworkFunctions::NEventReport (  
    const char * remote,  
    uint16_t portno,  
    const BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

12.214.2.6 NGet()

```
bool gdcmm::NormalizedNetworkFunctions::NGet (  
    const char * remote,  
    uint16_t portno,  
    const BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

12.214.2.7 NSet()

```
bool gdcmm::NormalizedNetworkFunctions::NSet (  
    const char * remote,  
    uint16_t portno,  
    const BaseQuery * query,  
    std::vector< DataSet > & retDataSets,  
    const char * aetitle,  
    const char * call) [static]
```

The documentation for this class was generated from the following file:

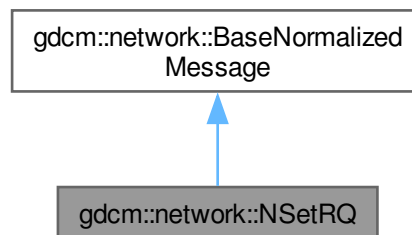
- [gdcmmNormalizedNetworkFunctions.h](#)

12.215 gdcn::network::NSetRQ Class Reference

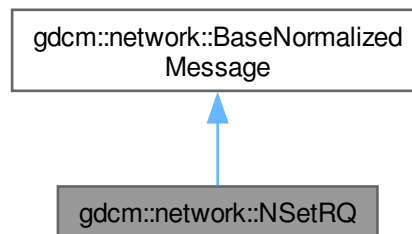
[NSetRQ](#).

```
#include <gdcnNSetMessages.h>
```

Inheritance diagram for gdcn::network::NSetRQ:



Collaboration diagram for gdcn::network::NSetRQ:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery) override

Public Member Functions inherited from [gdcn::network::BaseNormalizedMessage](#)

- virtual `~BaseNormalizedMessage ()`=default

12.215.1 Detailed Description

[NSetRQ](#).

this file defines the messages for the nset action

12.215.2 Member Function Documentation

12.215.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcmm::network::NSetRQ::ConstructPDV (  
    const ULConnection & inConnection,  
    const BaseQuery * inQuery)  [override], [virtual]
```

Implements [gdcmm::network::BaseNormalizedMessage](#).

The documentation for this class was generated from the following file:

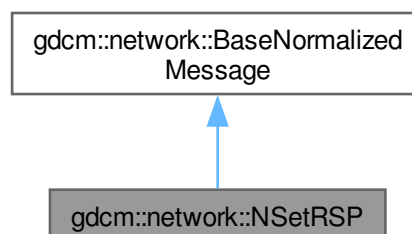
- [gdcmmNSetMessages.h](#)

12.216 gdcmm::network::NSetRSP Class Reference

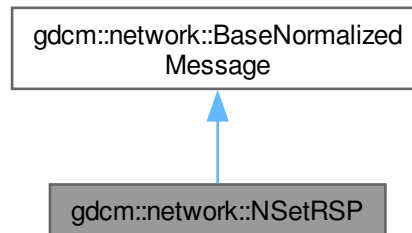
[NSetRSP](#) this file defines the messages for the nset action.

```
#include <gdcmmNSetMessages.h>
```

Inheritance diagram for gdcmm::network::NSetRSP:



Collaboration diagram for gdcm::network::NSetRSP:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

Public Member Functions inherited from [gdcm::network::BaseNormalizedMessage](#)

- `virtual ~BaseNormalizedMessage ()=default`
- `virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0`

12.216.1 Detailed Description

[NSetRSP](#) this file defines the messages for the nset action.

12.216.2 Member Function Documentation

12.216.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NSetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

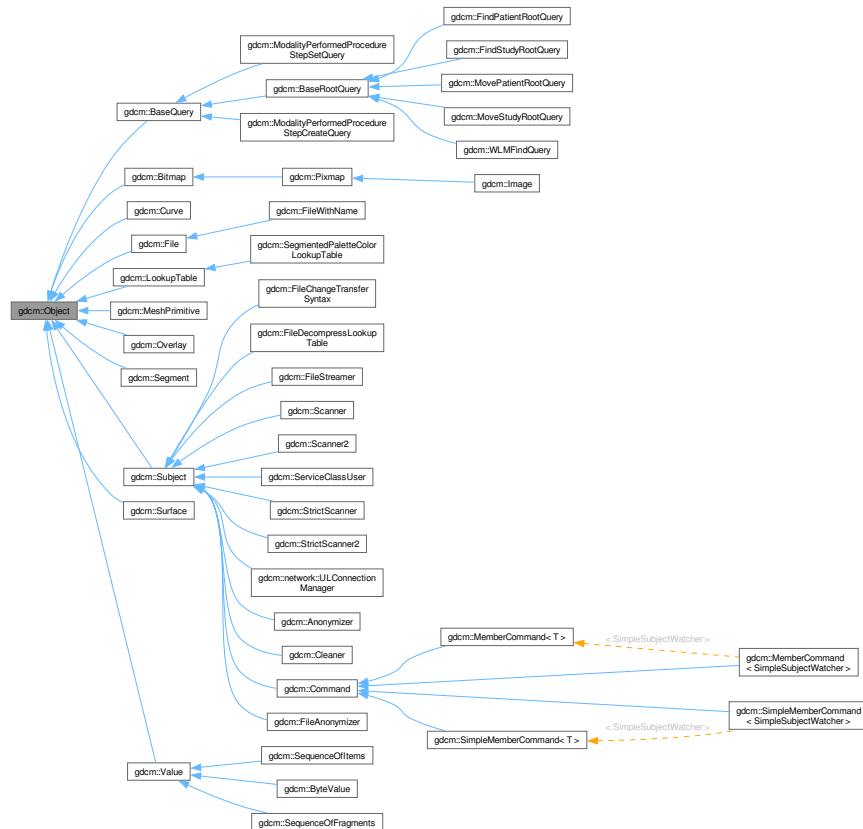
- [gdcmNSetMessages.h](#)

12.217 gdcm::Object Class Reference

[Object](#).

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



Public Member Functions

- [Object](#) ()
 - [Object](#) (const [Object](#) &)
 - virtual [~Object](#) ()
 - void [operator=](#) (const [Object](#) &)
 - virtual void [Print](#) (std::ostream &) const
- Special requirement for copy/cstor, assignment operator.

Protected Member Functions

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- `std::ostream & operator<< (std::ostream &os, const Object &obj)`
- `template<class ObjectType>
class SmartPointer`

12.217.1 Detailed Description

[Object](#).

Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

See also

[SmartPointer](#)

12.217.2 Constructor & Destructor Documentation

12.217.2.1 `Object()` [1/2]

`gdcm::Object::Object ()` [inline]

Referenced by [gdcm::LookupTable::LookupTable\(\)](#), [Object\(\)](#), [operator<<](#), [operator=\(\)](#), and [SmartPointer](#).

12.217.2.2 `~Object()`

`virtual gdcm::Object::~~Object ()` [inline], [virtual]

References [gdcm_forced_assert](#).

12.217.2.3 `Object()` [2/2]

`gdcm::Object::Object (
 const Object &)` [inline]

Special requirement for copy/cstor, assignment operator.

References [Object\(\)](#).

12.217.3 Member Function Documentation

12.217.3.1 operator=()

```
void gdcM::Object::operator= (  
    const Object & ) [inline]
```

References [Object\(\)](#).

12.217.3.2 Print()

```
virtual void gdcM::Object::Print (  
    std::ostream & ) const [inline], [virtual]
```

Reimplemented in [gdcM::BaseQuery](#), [gdcM::Bitmap](#), [gdcM::ByteValue](#), [gdcM::Curve](#), [gdcM::Image](#), [gdcM::LookupTable](#), [gdcM::Overlay](#), [gdcM::Pixmap](#), [gdcM::Scanner2](#), [gdcM::Scanner](#), [gdcM::SegmentedPaletteColorLookupTable](#), [gdcM::SequenceOfFragments](#), [gdcM::SequenceOfItems](#), [gdcM::StrictScanner2](#), and [gdcM::StrictScanner](#).

Examples

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by [gdcM::DataElement::operator<<](#), and [operator<<](#).

12.217.3.3 Register()

```
void gdcM::Object::Register () [inline], [protected]
```

References [gdcM_assert](#).

12.217.3.4 UnRegister()

```
void gdcM::Object::UnRegister () [inline], [protected]
```

References [gdcM_assert](#).

12.217.4 Friends And Related Symbol Documentation

12.217.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Object & obj) [friend]
```

References [Object\(\)](#), and [Print\(\)](#).

Referenced by [SmartPointer](#).

12.217.4.2 SmartPointer

```
template<class ObjectType>
friend class SmartPointer [friend]
```

References [Object\(\)](#), [operator<<](#), and [SmartPointer](#).

Referenced by [gdcm::Segment::AddSurface\(\)](#), [gdcm::Segment::GetSurface\(\)](#), [gdcm::Bitmap::SetLUT\(\)](#), [gdcm::Surface::SetMeshPrimitive\(\)](#), and [SmartPointer](#).

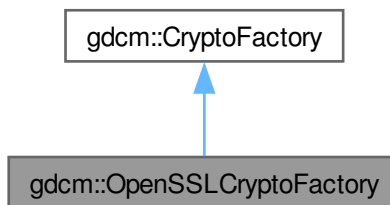
The documentation for this class was generated from the following file:

- [gdcmObject.h](#)

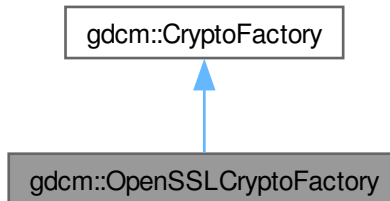
12.218 gdcm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLCryptoFactory:



Collaboration diagram for gdcm::OpenSSLCryptoFactory:



Public Member Functions

- [OpenSSLCryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSPProvider](#) ()

Protected Member Functions

- void [InitOpenSSL](#) ()

Protected Member Functions inherited from [gdcmm::CryptoFactory](#)

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

Additional Inherited Members

Public Types inherited from [gdcmm::CryptoFactory](#)

- enum [CryptoLib](#) {
 [DEFAULT](#) = 0 ,
 [OPENSSL](#) = 1 ,
 [CAPI](#) = 2 ,
 [OPENSSL7](#) = 3 }

Static Public Member Functions inherited from [gdcmm::CryptoFactory](#)

- static [CryptoFactory](#) * [GetFactoryInstance](#) ([CryptoLib](#) id=[DEFAULT](#))

12.218.1 Constructor & Destructor Documentation

12.218.1.1 [OpenSSLCryptoFactory](#)()

[gdcmm::OpenSSLCryptoFactory::OpenSSLCryptoFactory](#) (
 [CryptoLib](#) id) [inline]

References [gdcmm::CryptoFactory::CryptoFactory\(\)](#), and [gdcmmDebugMacro](#).

Referenced by [InitOpenSSL\(\)](#).

12.218.2 Member Function Documentation

12.218.2.1 CreateCMSProvider()

[CryptographicMessageSyntax](#) * gdcM::OpenSSLCryptoFactory::CreateCMSProvider () [inline], [virtual]

Implements [gdcM::CryptoFactory](#).

References [InitOpenSSL\(\)](#).

12.218.2.2 InitOpenSSL()

void gdcM::OpenSSLCryptoFactory::InitOpenSSL () [protected]

References [OpenSSLCryptoFactory\(\)](#).

Referenced by [CreateCMSProvider\(\)](#).

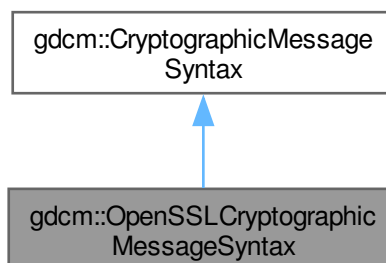
The documentation for this class was generated from the following file:

- [gdcMOpenSSLCryptoFactory.h](#)

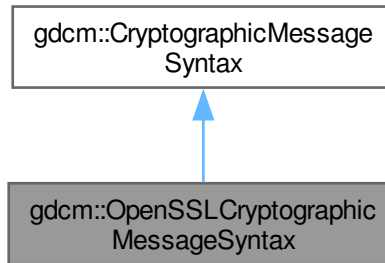
12.219 gdcM::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcMOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for `gdc::OpenSSLCryptographicMessageSyntax`:



Public Member Functions

- [OpenSSLCryptographicMessageSyntax](#) ()
- [~OpenSSLCryptographicMessageSyntax](#) ()
- `bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const`
decrypt content from a PKCS#7 envelopedData structure
- `bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const`
create a CMS envelopedData structure
- `CipherTypes GetCipherType () const`
- `bool ParseCertificateFile (const char *filename)`
- `bool ParseKeyFile (const char *filename)`
- `void SetCipherType (CipherTypes type)`
- `bool SetPassword (const char *pass, size_t passLen)`

Public Member Functions inherited from [gdc::CryptographicMessageSyntax](#)

- `CryptographicMessageSyntax ()=default`
- `CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete`
- `virtual ~CryptographicMessageSyntax ()=default`
- `void operator= (const CryptographicMessageSyntax &)=delete`

Additional Inherited Members

Public Types inherited from [gdc::CryptographicMessageSyntax](#)

- `enum CipherTypes {
 DES3_CIPHER ,
 AES128_CIPHER ,
 AES192_CIPHER ,
 AES256_CIPHER }`

12.219.1 Constructor & Destructor Documentation

12.219.1.1 OpenSSLCryptographicMessageSyntax()

gdcmm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ()

Referenced by [Decrypt\(\)](#).

12.219.1.2 ~OpenSSLCryptographicMessageSyntax()

gdcmm::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ()

12.219.2 Member Function Documentation

12.219.2.1 Decrypt()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::Decrypt (  
    char * output,  
    size_t & outlen,  
    const char * array,  
    size_t len) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

References [OpenSSLCryptographicMessageSyntax\(\)](#).

12.219.2.2 Encrypt()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::Encrypt (  
    char * output,  
    size_t & outlen,  
    const char * array,  
    size_t len) const [virtual]
```

create a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

12.219.2.3 GetCipherType()

[CipherTypes](#) gdcmm::OpenSSLCryptographicMessageSyntax::GetCipherType () const [virtual]

Implements [gdcmm::CryptographicMessageSyntax](#).

12.219.2.4 ParseCertificateFile()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (  
    const char * filename) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.219.2.5 ParseKeyFile()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (  
    const char * filename) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.219.2.6 SetCipherType()

```
void gdcmm::OpenSSLCryptographicMessageSyntax::SetCipherType (  
    CipherTypes type) [virtual]
```

Set Cipher [Type](#). Default is: AES256_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

12.219.2.7 SetPassword()

```
bool gdcmm::OpenSSLCryptographicMessageSyntax::SetPassword (  
    const char * pass,  
    size_t passLen) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

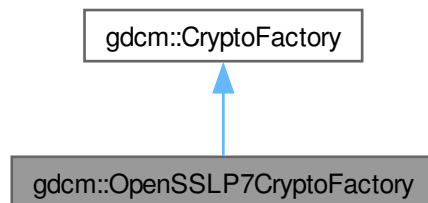
The documentation for this class was generated from the following file:

- [gdcmmOpenSSLCryptographicMessageSyntax.h](#)

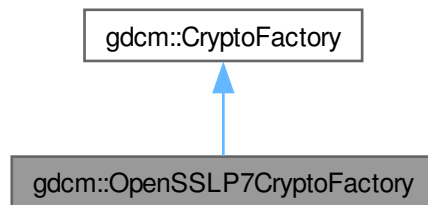
12.220 gdcM::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcMOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcM::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcM::OpenSSLP7CryptoFactory:



Public Member Functions

- [OpenSSLP7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) * [CreateCMSProvider](#) ()

Additional Inherited Members

Public Types inherited from [gdcM::CryptoFactory](#)

- enum [CryptoLib](#) {
 [DEFAULT](#) = 0 ,
 [OPENSSL](#) = 1 ,
 [CAPI](#) = 2 ,
 [OPENSSLP7](#) = 3 }

Static Public Member Functions inherited from [gdcM::CryptoFactory](#)

- static [CryptoFactory](#) * [GetFactoryInstance](#) ([CryptoLib](#) id=DEFAULT)

Protected Member Functions inherited from [gdcM::CryptoFactory](#)

- [CryptoFactory](#) ()=default
- [CryptoFactory](#) ([CryptoLib](#) id)
- [~CryptoFactory](#) ()=default

12.220.1 Constructor & Destructor Documentation

12.220.1.1 OpenSSLP7CryptoFactory()

[gdcM::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory](#) (
 [CryptoLib](#) id) [inline]

References [gdcM::CryptoFactory::CryptoFactory\(\)](#), and [gdcMDebugMacro](#).

12.220.2 Member Function Documentation

12.220.2.1 CreateCMSPProvider()

[CryptographicMessageSyntax](#) * [gdcM::OpenSSLP7CryptoFactory::CreateCMSPProvider](#) () [inline], [virtual]

Implements [gdcM::CryptoFactory](#).

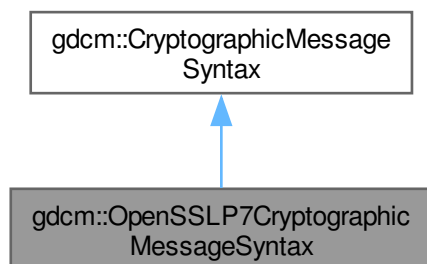
The documentation for this class was generated from the following file:

- [gdcMOpenSSLP7CryptoFactory.h](#)

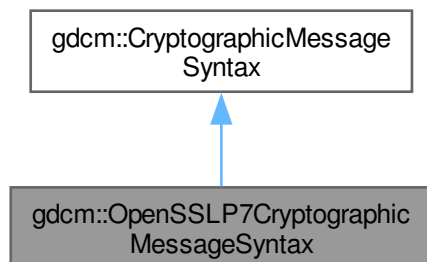
12.221 gdcM::OpenSSL7CryptographicMessageSyntax Class Reference

#include <gdcMOpenSSL7CryptographicMessageSyntax.h>

Inheritance diagram for gdcM::OpenSSL7CryptographicMessageSyntax:



Collaboration diagram for gdcM::OpenSSL7CryptographicMessageSyntax:



Public Member Functions

- [OpenSSL7CryptographicMessageSyntax](#) ()
- [~OpenSSL7CryptographicMessageSyntax](#) ()
- bool [Decrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
decrypt content from a PKCS#7 envelopedData structure
- bool [Encrypt](#) (char *output, size_t &outlen, const char *array, size_t len) const
create a PKCS#7 envelopedData structure
- [CipherTypes](#) [GetCipherType](#) () const
- bool [ParseCertificateFile](#) (const char *filename)
- bool [ParseKeyFile](#) (const char *filename)
- void [SetCipherType](#) (CipherTypes type)
- bool [SetPassword](#) (const char *, size_t)

Public Member Functions inherited from [gdcmm::CryptographicMessageSyntax](#)

- [CryptographicMessageSyntax](#) ()=default
- [CryptographicMessageSyntax](#) (const [CryptographicMessageSyntax](#) &)=delete
- virtual [~CryptographicMessageSyntax](#) ()=default
- void [operator=](#) (const [CryptographicMessageSyntax](#) &)=delete

Additional Inherited Members

Public Types inherited from [gdcmm::CryptographicMessageSyntax](#)

- enum [CipherTypes](#) {
[DES3_CIPHER](#) ,
[AES128_CIPHER](#) ,
[AES192_CIPHER](#) ,
[AES256_CIPHER](#) }

12.221.1 Detailed Description

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_↔ encrypt functionalities

See online documentation http://www.openssl.org/docs/crypto/PKCS7_encrypt.html

12.221.2 Constructor & Destructor Documentation

12.221.2.1 OpenSSLP7CryptographicMessageSyntax()

[gdcmm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax](#) ()

Referenced by [Decrypt\(\)](#).

12.221.2.2 ~OpenSSLP7CryptographicMessageSyntax()

[gdcmm::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax](#) ()

12.221.3 Member Function Documentation

12.221.3.1 Decrypt()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

References [OpenSSLP7CryptographicMessageSyntax\(\)](#).

12.221.3.2 Encrypt()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::Encrypt (  
    char * output,  
    size_t & outlen,  
    const char * array,  
    size_t len) const [virtual]
```

create a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

12.221.3.3 GetCipherType()

```
CipherTypes gdcmm::OpenSSLP7CryptographicMessageSyntax::GetCipherType () const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.221.3.4 ParseCertificateFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (  
    const char * filename) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.221.3.5 ParseKeyFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (  
    const char * filename) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

12.221.3.6 SetCipherType()

```
void gdcmm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (  
    CipherTypes type) [virtual]
```

Set Cipher [Type](#). Default is: AES256_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

12.221.3.7 SetPassword()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassword (
    const char * ,
    size_t ) [inline], [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

References [gdcmWarningMacro](#).

The documentation for this class was generated from the following file:

- [gdcmOpenSSLP7CryptographicMessageSyntax.h](#)

12.222 gdcm::Orientation Class Reference

class to handle [Orientation](#)

```
#include <gdcmOrientation.h>
```

Public Types

- enum [OrientationType](#) {
[UNKNOWN](#) ,
[AXIAL](#) ,
[CORONAL](#) ,
[SAGITTAL](#) ,
[OBLIQUE](#) }

Public Member Functions

- [Orientation](#) ()
- [~Orientation](#) ()=default
- void [Print](#) (std::ostream &) const
Print.

Static Public Member Functions

- static const char * [GetLabel](#) ([OrientationType](#) type)
Return the label of an [Orientation](#).
- static double [GetObliquityThresholdCosineValue](#) ()
- static [OrientationType](#) [GetType](#) (const double dircos[6])
- static void [SetObliquityThresholdCosineValue](#) (double val)
ObliquityThresholdCosineValue stuff.

Static Protected Member Functions

- static char [GetMajorAxisFromPatientRelativeDirectionCosine](#) (double x, double y, double z)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Orientation](#) &o)

12.222.1 Detailed Description

class to handle [Orientation](#)

12.222.2 Member Enumeration Documentation

12.222.2.1 OrientationType

enum [gdcmm::Orientation::OrientationType](#)

Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

Examples

[FixOrientation.cxx](#).

12.222.3 Constructor & Destructor Documentation

12.222.3.1 Orientation()

[gdcmm::Orientation::Orientation](#) ()

Referenced by [operator<<](#).

12.222.3.2 ~Orientation()

[gdcmm::Orientation::~~Orientation](#) () [default]

12.222.4 Member Function Documentation

12.222.4.1 GetLabel()

```
const char * gdcM::Orientation::GetLabel (
    OrientationType type) [static]
```

Return the label of an [Orientation](#).

Examples

[FixOrientation.cxx](#).

12.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
char gdcM::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
    double x,
    double y,
    double z) [static], [protected]
```

12.222.4.3 GetObliquityThresholdCosineValue()

```
double gdcM::Orientation::GetObliquityThresholdCosineValue () [static]
```

12.222.4.4 GetType()

```
OrientationType gdcM::Orientation::GetType (
    const double dircos[6]) [static]
```

Return the type of orientation from a direction cosines Input is an array of 6 double

Examples

[FixOrientation.cxx](#).

12.222.4.5 Print()

```
void gdcM::Orientation::Print (
    std::ostream & ) const
```

Print.

Referenced by [operator<<](#).

12.222.4.6 SetObliquityThresholdCosineValue()

```
void gdcm::Orientation::SetObliquityThresholdCosineValue (  
    double val) [static]
```

ObliquityThresholdCosineValue stuff.

12.222.5 Friends And Related Symbol Documentation

12.222.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const Orientation & o) [friend]
```

References [Orientation\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

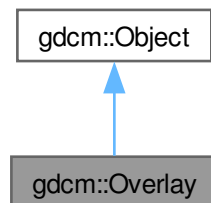
- [gdcmOrientation.h](#)

12.223 gdcm::Overlay Class Reference

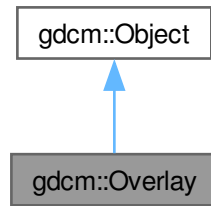
[Overlay](#) class.

```
#include <gdcmOverlay.h>
```

Inheritance diagram for gdcm::Overlay:



Collaboration diagram for `gdcm::Overlay`:



Public Types

- enum `OverlayType` {
`Invalid` = 0 ,
`Graphics` = 1 ,
`ROI` = 2 }

Public Member Functions

- `Overlay` ()
- `Overlay` (Overlay const &ov)
- `~Overlay` () override
- void `Decompress` (std::ostream &os) const
 Decode the internal OverlayData (packed bits) into unpacked representation.
- unsigned short `GetBitPosition` () const
 return bit position
- unsigned short `GetBitsAllocated` () const
 return bits allocated
- unsigned short `GetColumns` () const
 get columns
- const char * `GetDescription` () const
 get description
- unsigned short `GetGroup` () const
 Get Group number.
- const signed short * `GetOrigin` () const
 get origin
- const `ByteValue` & `GetOverlayData` () const
- unsigned short `GetRows` () const
 get rows
- const char * `GetType` () const
 get type
- `OverlayType` `GetTypeAsEnum` () const

- bool [GetUnpackBuffer](#) (char *buffer, size_t len) const
- size_t [GetUnpackBufferLength](#) () const
- bool [GrabOverlayFromPixelData](#) ([DataSet](#) const &ds)
- bool [IsEmpty](#) () const
 - Return whether or not the [Overlay](#) is empty:
- bool [IsInPixelData](#) () const
 - return if the [Overlay](#) is stored in the pixel data or not
- void [IsInPixelData](#) (bool b)
 - Set whether or no the OverlayData is in the Pixel Data:
- bool [IsZero](#) () const
 - return true if all bits are set to 0
- [Overlay](#) & [operator=](#) ([Overlay](#) const &ov)
- void [Print](#) (std::ostream &) const override
 - Print.
- void [SetBitPosition](#) (unsigned short bitposition)
 - set bit position
- void [SetBitsAllocated](#) (unsigned short bitsallocated)
 - set bits allocated
- void [SetColumns](#) (unsigned short columns)
 - set columns
- void [SetDescription](#) (const char *description)
 - set description
- void [SetFrameOrigin](#) (unsigned short frameorigin)
 - set frame origin
- void [SetGroup](#) (unsigned short group)
 - Set Group number.
- void [SetNumberOfFrames](#) (unsigned int numberofframes)
 - set number of frames
- void [SetOrigin](#) (const signed short origin[2])
 - set origin
- void [SetOverlay](#) (const char *array, size_t length)
 - set overlay from byte array + length
- void [SetRows](#) (unsigned short rows)
 - set rows
- void [SetType](#) (const char *type)
 - set type
- void [Update](#) (const [DataElement](#) &de)
 - Update overlay from data element de:

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
 - Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static const char * [GetOverlayTypeAsString](#) ([OverlayType](#) ot)
- static [OverlayType](#) [GetOverlayTypeFromString](#) (const char *)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.223.1 Detailed Description

[Overlay](#) class.

Note

see [AreOverlaysInPixelData](#)

[Todo](#) Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Example:

12.223.2 Member Enumeration Documentation

12.223.2.1 [OverlayType](#)

enum [gdcm::Overlay::OverlayType](#)

Enumerator

Invalid	
Graphics	
ROI	

12.223.3 Constructor & Destructor Documentation

12.223.3.1 [Overlay\(\)](#) [1/2]

[gdcm::Overlay::Overlay](#) ()

Referenced by [Overlay\(\)](#), and [operator=\(\)](#).

12.223.3.2 ~Overlay()

`gdcmm::Overlay::~~Overlay () [override]`

12.223.3.3 Overlay() [2/2]

`gdcmm::Overlay::Overlay (
Overlay const & ov)`

References [Overlay\(\)](#).

12.223.4 Member Function Documentation

12.223.4.1 Decompress()

`void gdcmm::Overlay::Decompress (
std::ostream & os) const`

Decode the internal OverlayData (packed bits) into unpacked representation.

12.223.4.2 GetBitPosition()

`unsigned short gdcmm::Overlay::GetBitPosition () const`

return bit position

12.223.4.3 GetBitsAllocated()

`unsigned short gdcmm::Overlay::GetBitsAllocated () const`

return bits allocated

12.223.4.4 GetColumns()

`unsigned short gdcmm::Overlay::GetColumns () const`

get columns

12.223.4.5 GetDescription()

`const char * gdcmm::Overlay::GetDescription () const`

get description

12.223.4.6 GetGroup()

unsigned short gdcm::Overlay::GetGroup () const

Get Group number.

12.223.4.7 GetOrigin()

const signed short * gdcm::Overlay::GetOrigin () const

get origin

12.223.4.8 GetOverlayData()

const [ByteValue](#) & gdcm::Overlay::GetOverlayData () const

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

12.223.4.9 GetOverlayTypeAsString()

const char * gdcm::Overlay::GetOverlayTypeAsString (
 [OverlayType](#) ot) [static]

12.223.4.10 GetOverlayTypeFromString()

[OverlayType](#) gdcm::Overlay::GetOverlayTypeFromString (
 const char *) [static]

12.223.4.11 GetRows()

unsigned short gdcm::Overlay::GetRows () const

get rows

12.223.4.12 GetType()

const char * gdcm::Overlay::GetType () const

get type

12.223.4.13 GetTypeAsEnum()

[OverlayType](#) gdcm::Overlay::GetTypeAsEnum () const

12.223.4.14 GetUnpackBuffer()

```
bool gdcm::Overlay::GetUnpackBuffer (
    char * buffer,
    size_t len) const
```

Retrieve the unpack buffer for [Overlay](#). This is an error if the size is below [GetUnpackBufferLength\(\)](#)

12.223.4.15 GetUnpackBufferLength()

```
size_t gdcm::Overlay::GetUnpackBufferLength () const
```

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

12.223.4.16 GrabOverlayFromPixelData()

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
    DataSet const & ds)
```

Warning

Before calling this method, you must verify the consistency between the image metadata ([Image PixelFormat](#), Rows, Columns) and the overlay parameters. This pre-verification is required to ensure that the bit-depth is compatible and that the overlay data fits within the allocated pixel storage.

12.223.4.17 IsEmpty()

```
bool gdcm::Overlay::IsEmpty () const
```

Return whether or not the [Overlay](#) is empty:

12.223.4.18 IsInPixelData() [1/2]

```
bool gdcm::Overlay::IsInPixelData () const
```

return if the [Overlay](#) is stored in the pixel data or not

12.223.4.19 IsInPixelData() [2/2]

```
void gdcm::Overlay::IsInPixelData (
    bool b)
```

Set whether or no the OverlayData is in the Pixel Data:

12.223.4.20 IsZero()

```
bool gdcmm::Overlay::IsZero () const
```

return true if all bits are set to 0

12.223.4.21 operator=()

```
Overlay & gdcmm::Overlay::operator= (  
    Overlay const & ov)
```

References [Overlay\(\)](#).

12.223.4.22 Print()

```
void gdcmm::Overlay::Print (  
    std::ostream & ) const    [override], [virtual]
```

Print.

Reimplemented from [gdcmm::Object](#).

12.223.4.23 SetBitPosition()

```
void gdcmm::Overlay::SetBitPosition (  
    unsigned short bitposition)
```

set bit position

12.223.4.24 SetBitsAllocated()

```
void gdcmm::Overlay::SetBitsAllocated (  
    unsigned short bitsallocated)
```

set bits allocated

12.223.4.25 SetColumns()

```
void gdcmm::Overlay::SetColumns (  
    unsigned short columns)
```

set columns

12.223.4.26 SetDescription()

```
void gdcm::Overlay::SetDescription (  
    const char * description)
```

set description

12.223.4.27 SetFrameOrigin()

```
void gdcm::Overlay::SetFrameOrigin (  
    unsigned short frameorigin)
```

set frame origin

12.223.4.28 SetGroup()

```
void gdcm::Overlay::SetGroup (  
    unsigned short group)
```

Set Group number.

12.223.4.29 SetNumberOfFrames()

```
void gdcm::Overlay::SetNumberOfFrames (  
    unsigned int numberofframes)
```

set number of frames

12.223.4.30 SetOrigin()

```
void gdcm::Overlay::SetOrigin (  
    const signed short origin[2])
```

set origin

12.223.4.31 SetOverlay()

```
void gdcm::Overlay::SetOverlay (  
    const char * array,  
    size_t length)
```

set overlay from byte array + length

12.223.4.32 SetRows()

```
void gdcm::Overlay::SetRows (  
    unsigned short rows)
```

set rows

12.223.4.33 SetType()

```
void gdcm::Overlay::SetType (  
    const char * type)
```

set type

12.223.4.34 Update()

```
void gdcm::Overlay::Update (  
    const DataElement & de)
```

Update overlay from data element de:

The documentation for this class was generated from the following file:

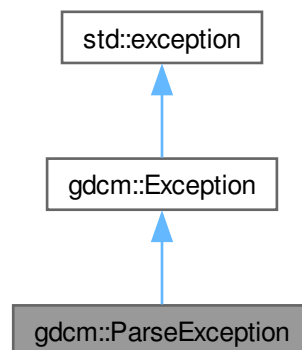
- [gdcmOverlay.h](#)

12.224 gdcm::ParseException Class Reference

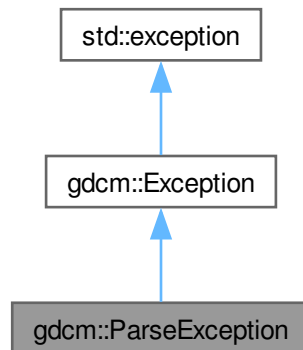
[ParseException](#) Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- [ParseException](#) ()=default
- [ParseException](#) (const [ParseException](#) &orig)
- [~ParseException](#) () override throw ()
- const [DataElement](#) & [GetLastElement](#) () const
- [ParseException](#) & [operator=](#) (const [ParseException](#) &orig)
- void [SetLastElement](#) (const [DataElement](#) &de)

Public Member Functions inherited from [gdcm::Exception](#)

- [Exception](#) (const char *desc="None", const char *file=__FILE__, unsigned int lineNumber=__LINE__, const char *func=__FUNCTION__)
- [~Exception](#) () override throw ()
- const char * [GetDescription](#) () const
Return the Description.
- const char * [what](#) () const override throw ()
what implementation

12.224.1 Detailed Description

[ParseException](#) Standard exception handling object.

12.224.2 Constructor & Destructor Documentation

12.224.2.1 [ParseException](#)() [1/2]

`gdcm::ParseException::ParseException ()` [default]

Referenced by [ParseException\(\)](#), and [operator=\(\)](#).

12.224.2.2 ~ParseException()

`gdcm::ParseException::~~ParseException () throw ()` [inline], [override]

12.224.2.3 ParseException() [2/2]

`gdcm::ParseException::ParseException (`
 `const ParseException & orig)` [inline]

References [gdcm::Exception::Exception\(\)](#), and [ParseException\(\)](#).

12.224.3 Member Function Documentation

12.224.3.1 GetLastElement()

`const DataElement & gdcm::ParseException::GetLastElement () const` [inline]

12.224.3.2 operator=()

`ParseException & gdcm::ParseException::operator= (`
 `const ParseException & orig)` [inline]

Assignment operator.

References [ParseException\(\)](#).

12.224.3.3 SetLastElement()

`void gdcm::ParseException::SetLastElement (`
 `const DataElement & de)` [inline]

Equivalence operator.

Referenced by [gdcm::BasicOffsetTable::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), and [gdcm::Fragment::ReadValue\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmParseException.h](#)

12.225 gdcm::Parser Class Reference

[Parser](#) ala XML_Parser from expat (SAX).

```
#include <gdcmParser.h>
```

Public Types

- typedef void(* [EndElementHandler](#)) (void *userData, const [Tag](#) &name)
- enum [ErrorType](#) {
 [NoError](#) ,
 [NoMemoryError](#) ,
 [SyntaxError](#) ,
 [NoElementsError](#) ,
 [TagMismatchError](#) ,
 [DuplicateAttributeError](#) ,
 [JunkAfterDocElementError](#) ,
 [UndefinedEntityError](#) ,
 [UnexpectedStateError](#) }
- typedef void(* [StartElementHandler](#)) (void *userData, const [Tag](#) &tag, const char *atts[])

Public Member Functions

- [Parser](#) ()
- [~Parser](#) ()
- unsigned long [GetCurrentByteIndex](#) () const
- [ErrorType](#) [GetErrorCode](#) () const
- void * [GetUserData](#) () const
- bool [Parse](#) (const char *s, int len, bool isFinal)
- void [SetElementHandler](#) ([StartElementHandler](#) start, [EndElementHandler](#) end)
- void [SetUserData](#) (void *userData)

Static Public Member Functions

- static const char * [GetErrorString](#) ([ErrorType](#) const &err)

Protected Member Functions

- char * [GetBuffer](#) (int len)
- bool [ParseBuffer](#) (int len, bool isFinal)
- [ErrorType](#) [Process](#) ()

12.225.1 Detailed Description

[Parser](#) ala XML_Parser from expat (SAX).

Detailed description here

Note

Simple API for DICOM

12.225.2 Member Typedef Documentation

12.225.2.1 EndElementHandler

typedef void(* gdcmm::Parser::EndElementHandler) (void *userData, const [Tag](#) &name)

12.225.2.2 StartElementHandler

typedef void(* gdcmm::Parser::StartElementHandler) (void *userData, const [Tag](#) &tag, const char *atts[])

12.225.3 Member Enumeration Documentation

12.225.3.1 ErrorType

enum [gdcmm::Parser::ErrorType](#)

Enumerator

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

12.225.4 Constructor & Destructor Documentation

12.225.4.1 Parser()

[gdcmm::Parser::Parser](#) () [inline]

References [NoError](#).

12.225.4.2 ~Parser()

[gdcmm::Parser::~~Parser](#) () [inline]

12.225.5 Member Function Documentation

12.225.5.1 GetBuffer()

```
char * gdcm::Parser::GetBuffer (  
    int len)    [protected]
```

12.225.5.2 GetCurrentByteIndex()

```
unsigned long gdcm::Parser::GetCurrentByteIndex () const
```

12.225.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode () const
```

12.225.5.4 GetErrorString()

```
const char * gdcm::Parser::GetErrorString (  
    ErrorType const & err)    [static]
```

12.225.5.5 GetUserData()

```
void * gdcm::Parser::GetUserData () const
```

12.225.5.6 Parse()

```
bool gdcm::Parser::Parse (  
    const char * s,  
    int len,  
    bool isFinal)
```

12.225.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (  
    int len,  
    bool isFinal)    [protected]
```

12.225.5.8 Process()

```
ErrorType gdcm::Parser::Process ()    [protected]
```

12.225.5.9 SetElementHandler()

```
void gdcM::Parser::SetElementHandler (
    StartElementHandler start,
    EndElementHandler end)
```

12.225.5.10 SetUserData()

```
void gdcM::Parser::SetUserData (
    void * userData)
```

The documentation for this class was generated from the following file:

- [gdcMParser.h](#)

12.226 gdcM::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcMPatient.h>
```

Public Member Functions

- [Patient](#) ()=default

12.226.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

12.226.2 Constructor & Destructor Documentation

12.226.2.1 Patient()

```
gdcM::Patient::Patient () [default]
```

The documentation for this class was generated from the following file:

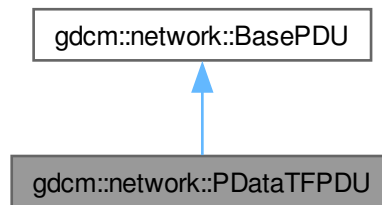
- [gdcMPatient.h](#)

12.227 gdcn::network::PDataTFPDU Class Reference

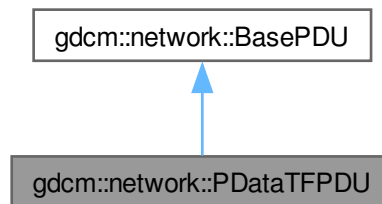
[PDataTFPDU](#).

```
#include <gdcnPDataTFPDU.h>
```

Inheritance diagram for gdcn::network::PDataTFPDU:



Collaboration diagram for gdcn::network::PDataTFPDU:



Public Types

- typedef std::vector< [PresentationDataValue](#) >::size_type [SizeType](#)

Public Member Functions

- [PDataTFPDU](#) ()
- void [AddPresentationDataValue](#) ([PresentationDataValue](#) const &pdv)
- [SizeType](#) [GetNumberOfPresentationDataValues](#) () const
- [PresentationDataValue](#) const & [GetPresentationDataValue](#) ([SizeType](#) i) const
- bool [IsLastFragment](#) () const override
- void [Print](#) (std::ostream &os) const override
- std::istream & [Read](#) (std::istream &is) override
- size_t [Size](#) () const override
- const std::ostream & [Write](#) (std::ostream &os) const override

Public Member Functions inherited from [gdcn::network::BasePDU](#)

- virtual [~BasePDU](#) ()=default

Protected Member Functions

- std::istream & [ReadInto](#) (std::istream &is, std::ostream &os)

12.227.1 Detailed Description

[PDataTFPDU](#).

[Table 9-22 P-DATA-TF PDU FIELDS](#)

12.227.2 Member Typedef Documentation

12.227.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcn::network::PDataTFPDU::SizeType
```

12.227.3 Constructor & Destructor Documentation

12.227.3.1 PDataTFPDU()

```
gdcn::network::PDataTFPDU::PDataTFPDU ()
```

12.227.4 Member Function Documentation

12.227.4.1 AddPresentationDataValue()

```
void gdcn::network::PDataTFPDU::AddPresentationDataValue (  
    PresentationDataValue const & pdv) [inline]
```

References [gdcn_assert](#), and [Size\(\)](#).

12.227.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcn::network::PDataTFPDU::GetNumberOfPresentationDataValues () const [inline]
```


12.227.4.3 GetPresentationDataValue()

[PresentationDataValue](#) const & gdcm::network::PDataTFPDU::GetPresentationDataValue (
 [SizeType](#) i) const [inline]

References [gdcm_assert](#).

12.227.4.4 IsLastFragment()

bool gdcm::network::PDataTFPDU::IsLastFragment () const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.227.4.5 Print()

void gdcm::network::PDataTFPDU::Print (
 std::ostream & os) const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.227.4.6 Read()

std::istream & gdcm::network::PDataTFPDU::Read (
 std::istream & is) [override], [virtual]

Implements [gdcm::network::BasePDU](#).

12.227.4.7 ReadInto()

std::istream & gdcm::network::PDataTFPDU::ReadInto (
 std::istream & is,
 std::ostream & os) [protected]

12.227.4.8 Size()

size_t gdcm::network::PDataTFPDU::Size () const [override], [virtual]

Implements [gdcm::network::BasePDU](#).

Referenced by [AddPresentationDataValue\(\)](#).

12.227.4.9 Write()

```
const std::ostream & gdcm::network::PDataTFPDU::Write (
    std::ostream & os) const    [override], [virtual]
```

Implements [gdcm::network::BasePDU](#).

The documentation for this class was generated from the following file:

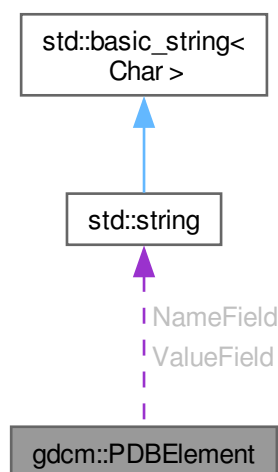
- [gdcmPDataTFPDU.h](#)

12.228 gdcm::PDBelement Class Reference

Class to represent a PDB [Element](#).

```
#include <gdcmPDBelement.h>
```

Collaboration diagram for gdcm::PDBelement:



Public Member Functions

- [PDBelement](#) ()=default
- const char * [GetName](#) () const
Set/Get Name.
- const char * [GetValue](#) () const
Set/Get Value.
- bool [operator==](#) (const [PDBelement](#) &de) const
- void [SetName](#) (const char *name)
- void [SetValue](#) (const char *value)

Protected Attributes

- std::string [NameField](#)
- std::string [ValueField](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [PDBelement](#) &val)

12.228.1 Detailed Description

Class to represent a PDB [Element](#).

See also

[PDBeheader](#)

12.228.2 Constructor & Destructor Documentation

12.228.2.1 PDBelement()

`gdcmm::PDBelement::PDBelement ()` [default]

References [PDBelement\(\)](#), and [operator<<](#).

Referenced by [PDBelement\(\)](#), [operator<<](#), and [operator==\(\)](#).

12.228.3 Member Function Documentation

12.228.3.1 GetName()

`const char * gdcmm::PDBelement::GetName () const` [inline]

Set/Get Name.

References [NameField](#).

12.228.3.2 GetValue()

`const char * gdcmm::PDBelement::GetValue () const` [inline]

Set/Get [Value](#).

References [ValueField](#).

12.228.3.3 operator==()

```
bool gdcmm::PDBelement::operator==(
    const PDBelement & de) const    [inline]
```

References [PDBelement\(\)](#), [NameField](#), and [ValueField](#).

12.228.3.4 setName()

```
void gdcmm::PDBelement::SetName (
    const char * name)    [inline]
```

References [NameField](#).

12.228.3.5 SetValue()

```
void gdcmm::PDBelement::SetValue (
    const char * value)    [inline]
```

References [ValueField](#).

12.228.4 Friends And Related Symbol Documentation

12.228.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const PDBelement & val)    [friend]
```

References [PDBelement\(\)](#), [NameField](#), and [ValueField](#).

Referenced by [PDBelement\(\)](#).

12.228.5 Member Data Documentation

12.228.5.1 NameField

```
std::string gdcmm::PDBelement::NameField    [protected]
```

Referenced by [GetName\(\)](#), [operator<<](#), [operator==\(\)](#), and [SetName\(\)](#).

12.228.5.2 ValueField

std::string gdcm::PDBElement::ValueField [protected]

Referenced by [GetValue\(\)](#), [operator<<](#), [operator==\(\(\)\)](#), and [SetValue\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmPDBElement.h](#)

12.229 gdcm::PDBHeader Class Reference

Class for [PDBHeader](#).

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- [PDBHeader](#) ()=default
- [~PDBHeader](#) ()=default
- bool [FindPDBElementByName](#) (const char *name)
Return true if the PDB element matching name is found or not.
- const [PDBElement](#) & [GetPDBElementByName](#) (const char *name)
- bool [LoadFromDataElement](#) ([DataElement](#) const &de)
Load the PDB Header from a [DataElement](#) of a [DataSet](#).
- void [Print](#) (std::ostream &os) const
Print.

Static Public Member Functions

- static const [PrivateTag](#) & [GetPDBInfoTag](#) ()
Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

Protected Member Functions

- const [PDBElement](#) & [GetPDBEEnd](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [PDBHeader](#) &d)

12.229.1 Detailed Description

Class for [PDBHeader](#).

GEMS MR [Image](#) have an [Attribute](#) (0025,1b,GEMS_SERS_01) which store the Acquisition parameter of the MR [Image](#). It is compressed and can therefore not be used as is. This class de-encapsulated the Protocol Data Block and allow users to query element by name.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

: the API of this class might change.

: SEDESC is not always pure ASCII it can contains latin1

See also

[CSAHeader](#)

12.229.2 Constructor & Destructor Documentation

12.229.2.1 PDBHeader()

gdcm::PDBHeader::PDBHeader () [default]

Referenced by [operator<<](#).

12.229.2.2 ~PDBHeader()

gdcm::PDBHeader::~~PDBHeader () [default]

12.229.3 Member Function Documentation

12.229.3.1 FindPDBelementByName()

```
bool gdcm::PDBHeader::FindPDBelementByName (
    const char * name)
```

Return true if the PDB element matching name is found or not.

12.229.3.2 GetPDBeEnd()

```
const PDBelement & gdcm::PDBHeader::GetPDBeEnd () const [protected]
```

12.229.3.3 GetPDBelementByName()

```
const PDBelement & gdcm::PDBHeader::GetPDBelementByName (  
    const char * name)
```

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

12.229.3.4 GetPDBInfoTag()

```
const PrivateTag & gdcm::PDBHeader::GetPDBInfoTag () [static]
```

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

12.229.3.5 LoadFromDataElement()

```
bool gdcm::PDBHeader::LoadFromDataElement (  
    DataElement const & de)
```

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

12.229.3.6 Print()

```
void gdcm::PDBHeader::Print (  
    std::ostream & os) const
```

Print.

Referenced by [operator<<](#).

12.229.4 Friends And Related Symbol Documentation

12.229.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const PDBHeader & d) [friend]
```

References [PDBHeader\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

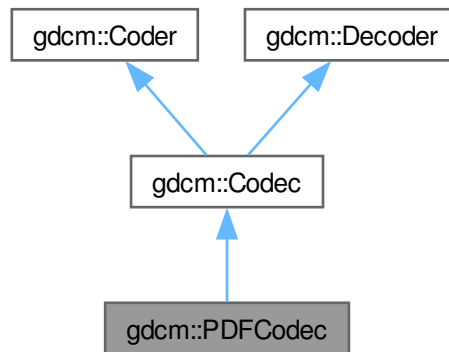
- [gdcmPDBHeader.h](#)

12.230 gdcmm::PDFCodec Class Reference

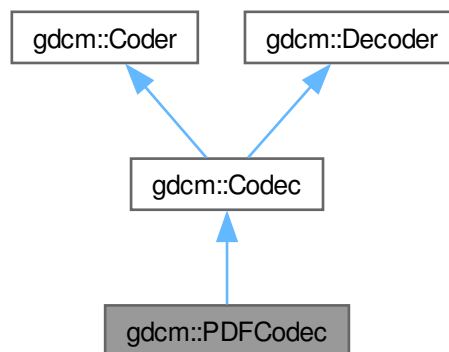
[PDFCodec](#) class.

```
#include <gdcmmPDFCodec.h>
```

Inheritance diagram for gdcmm::PDFCodec:



Collaboration diagram for gdcmm::PDFCodec:



Public Member Functions

- [PDFCodec](#) ()
- [~PDFCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &) const override
Return whether this decoder support this transfer syntax (can decode it).
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from [gdcm::Decoder](#)

- virtual bool [DecodeByStreams](#) (std::istream &, std::ostream &)

12.230.1 Detailed Description

[PDFCodec](#) class.

12.230.2 Constructor & Destructor Documentation

12.230.2.1 [PDFCodec](#)()

[gdcm::PDFCodec::PDFCodec](#) ()

12.230.2.2 ~PDFCodec()

`gdcmm::PDFCodec::~~PDFCodec ()` [override]

12.230.3 Member Function Documentation

12.230.3.1 CanCode()

`bool gdcmm::PDFCodec::CanCode (`
 [TransferSyntax](#) const &) const [inline], [override], [virtual]

Return whether this coder support this transfer syntax (can code it).

Implements [gdcmm::Coder](#).

12.230.3.2 CanDecode()

`bool gdcmm::PDFCodec::CanDecode (`
 [TransferSyntax](#) const &) const [inline], [override], [virtual]

Return whether this decoder support this transfer syntax (can decode it).

Implements [gdcmm::Decoder](#).

12.230.3.3 Decode()

`bool gdcmm::PDFCodec::Decode (`
 [DataElement](#) const & ,
 [DataElement](#) &) [override], [virtual]

Decode.

Reimplemented from [gdcmm::Decoder](#).

The documentation for this class was generated from the following file:

- [gdcmmPDFCodec.h](#)

12.231 gdcmm::network::PDUFactory Class Reference

[PDUFactory](#) basically, given an initial byte, construct the.

```
#include <gdcmmPDUFactory.h>
```

Static Public Member Functions

- static [BasePDU](#) * [ConstructAbortPDU](#) ()
- static [BasePDU](#) * [ConstructPDU](#) (uint8_t itemtype)
- static [BasePDU](#) * [ConstructReleasePDU](#) ()
- static std::vector< [BasePDU](#) * > [CreateCEchoPDU](#) (const [ULConnection](#) &inConnection)
- static std::vector< [BasePDU](#) * > [CreateCFindPDU](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [BasePDU](#) * > [CreateCMovePDU](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)
- static std::vector< [BasePDU](#) * > [CreateCStoreRQPDU](#) (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [BasePDU](#) * > [CreateCStoreRSPDU](#) (const [DataSet](#) *inDataSet, const [BasePDU](#) *inPC)
- static std::vector< [BasePDU](#) * > [CreateNActionPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [BasePDU](#) * > [CreateNCreatePDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [BasePDU](#) * > [CreateNDeletePDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [BasePDU](#) * > [CreateNEventReportPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [BasePDU](#) * > [CreateNGetPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static std::vector< [BasePDU](#) * > [CreateNSetPDU](#) (const [ULConnection](#) &inConnection, const [BaseQuery](#) *inQuery)
- static [EEventID](#) [DetermineEventByPDU](#) (const [BasePDU](#) *inPDU)
- static std::vector< [PresentationDataValue](#) > [GetPDVs](#) (const std::vector< [BasePDU](#) * > &inData← PDUs)

12.231.1 Detailed Description

[PDUFactory](#) basically, given an initial byte, construct the.

appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

12.231.2 Member Function Documentation

12.231.2.1 ConstructAbortPDU()

[BasePDU](#) * gdcn::network::PDUFactory::ConstructAbortPDU () [static]

12.231.2.2 ConstructPDU()

[BasePDU](#) * gdcn::network::PDUFactory::ConstructPDU (uint8_t itemtype) [static]

12.231.2.3 ConstructReleasePDU()

```
BasePDU * gdcm::network::PDUFactory::ConstructReleasePDU () [static]
```

12.231.2.4 CreateCEchoPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCEchoPDU (
    const ULConnection & inConnection) [static]
```

12.231.2.5 CreateCFindPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCFindPDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery) [static]
```

12.231.2.6 CreateCMovePDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCMovePDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery) [static]
```

12.231.2.7 CreateCStoreRQPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRQPDU (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true) [static]
```

12.231.2.8 CreateCStoreRSPPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRSPPDU (
    const DataSet * inDataSet,
    const BasePDU * inPC) [static]
```

12.231.2.9 CreateNActionPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNActionPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.10 CreateNCreatePDU()

```
std::vector< BasePDU * > gdcmm::network::PDUFactory::CreateNCreatePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.11 CreateNDeletePDU()

```
std::vector< BasePDU * > gdcmm::network::PDUFactory::CreateNDeletePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.12 CreateNEventReportPDU()

```
std::vector< BasePDU * > gdcmm::network::PDUFactory::CreateNEventReportPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.13 CreateNGetPDU()

```
std::vector< BasePDU * > gdcmm::network::PDUFactory::CreateNGetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.14 CreateNSetPDU()

```
std::vector< BasePDU * > gdcmm::network::PDUFactory::CreateNSetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery) [static]
```

12.231.2.15 DetermineEventByPDU()

```
EEEventID gdcmm::network::PDUFactory::DetermineEventByPDU (
    const BasePDU * inPDU) [static]
```

12.231.2.16 GetPDVs()

```
std::vector< PresentationDataValue > gdcmm::network::PDUFactory::GetPDVs (
    const std::vector< BasePDU * > & inDataPDUs) [static]
```

The documentation for this class was generated from the following file:

- [gdcmmPDUFactory.h](#)

12.232 gdcm::PersonName Class Reference

[PersonName](#) class.

```
#include <gdcmPersonName.h>
```

Public Member Functions

- unsigned int [GetMaxLength](#) () const
- unsigned int [GetNumberOfComponents](#) () const
- void [Print](#) (std::ostream &os) const
- void [SetBlob](#) (const std::vector< char > &v)
- void [SetComponents](#) (const char *comp1="", const char *comp2="", const char *comp3="", const char *comp4="", const char *comp5="")
- void [SetComponents](#) (const char *components[])

Public Attributes

- char [Component](#) [[MaxNumberOfComponents](#)][[MaxLength](#)+1]

Static Public Attributes

- static const unsigned int [MaxLength](#) = 64
- static const unsigned int [MaxNumberOfComponents](#) = 5
- static const char [Padding](#) = ' '
- static const char [Separator](#) = '^'

12.232.1 Detailed Description

[PersonName](#) class.

12.232.2 Member Function Documentation

12.232.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength () const [inline]
```

References [MaxLength](#).

Referenced by [SetComponents\(\)](#).

12.232.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents () const [inline]
```

References [Component](#).

12.232.2.3 Print()

```
void gdcm::PersonName::Print (  
    std::ostream & os) const    [inline]
```

References [Component](#).

12.232.2.4 SetBlob()

```
void gdcm::PersonName::SetBlob (  
    const std::vector< char > & v)    [inline]
```

12.232.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (  
    const char * comp1 = "",  
    const char * comp2 = "",  
    const char * comp3 = "",  
    const char * comp4 = "",  
    const char * comp5 = "")    [inline]
```

References [SetComponents\(\)](#).

Referenced by [SetComponents\(\)](#).

12.232.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (  
    const char * components[])    [inline]
```

References [Component](#), [gdcm_assert](#), and [GetMaxLength\(\)](#).

12.232.3 Member Data Documentation

12.232.3.1 Component

```
char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]
```

Referenced by [GetNumberOfComponents\(\)](#), [Print\(\)](#), and [SetComponents\(\)](#).

12.232.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64    [static]
```

Referenced by [GetMaxLength\(\)](#).

12.232.3.3 MaxNumberOfComponents

```
const unsigned int gdcM::PersonName::MaxNumberOfComponents = 5 [static]
```

12.232.3.4 Padding

```
const char gdcM::PersonName::Padding = ' ' [static]
```

12.232.3.5 Separator

```
const char gdcM::PersonName::Separator = '^' [static]
```

The documentation for this class was generated from the following file:

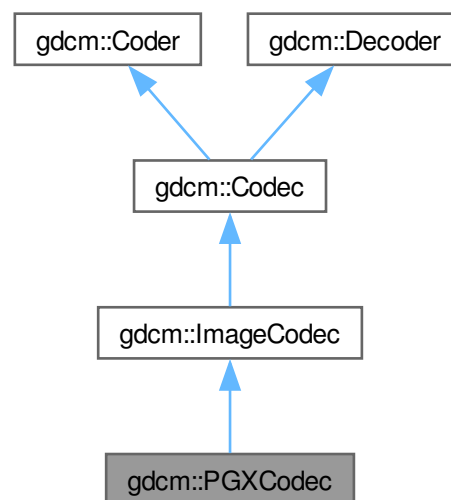
- [gdcMPersonName.h](#)

12.233 gdcM::PGXCodec Class Reference

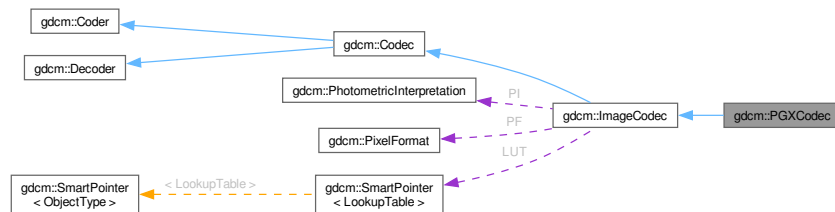
Class to do PGX.

```
#include <gdcMPGXCodec.h>
```

Inheritance diagram for gdcM::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



Public Member Functions

- [PGXCodec](#) ()
- [~PGXCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [Read](#) (const char *filename, [DataElement](#) &out) const
- bool [Write](#) (const char *filename, const [DataElement](#) &out) const

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- bool [Decode](#) ([DataElement](#) const &is_, [DataElement](#) &os) override
Decode.
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_) Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > LUTPtr

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.233.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

12.233.2 Constructor & Destructor Documentation

12.233.2.1 PGXCodec()

```
gdcmm::PGXCodec::PGXCodec ()
```

12.233.2.2 ~PGXCodec()

```
gdcmm::PGXCodec::~PGXCodec () [override]
```

12.233.3 Member Function Documentation

12.233.3.1 CanCode()

```
bool gdcmm::PGXCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcmm::ImageCodec](#).

12.233.3.2 CanDecode()

```
bool gdcmm::PGXCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcmm::ImageCodec](#).

12.233.3.3 Clone()

```
ImageCodec * gdcmm::PGXCodec::Clone () const [override], [virtual]
```

Implements [gdcmm::ImageCodec](#).

References [gdcmm::ImageCodec::ImageCodec\(\)](#).

12.233.3.4 GetHeaderInfo()

```
bool gdcM::PGXCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts) [override], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

12.233.3.5 Read()

```
bool gdcM::PGXCodec::Read (
    const char * filename,
    DataElement & out) const
```

12.233.3.6 Write()

```
bool gdcM::PGXCodec::Write (
    const char * filename,
    const DataElement & out) const
```

The documentation for this class was generated from the following file:

- [gdcM_PGXCodec.h](#)

12.234 gdcM::PhotometricInterpretation Class Reference

Class to represent an [PhotometricInterpretation](#).

```
#include <gdcMPhotometricInterpretation.h>
```

Public Types

- enum [PIType](#) {
[UNKNOWN](#) = 0 ,
[MONOCHROME1](#) ,
[MONOCHROME2](#) ,
[PALETTE_COLOR](#) ,
[RGB](#) ,
[HSV](#) ,
[ARGB](#) ,
[CMYK](#) ,
[YBR_FULL](#) ,
[YBR_FULL_422](#) ,
[YBR_PARTIAL_422](#) ,
[YBR_PARTIAL_420](#) ,
[YBR_ICT](#) ,
[YBR_RCT](#) ,
[PI_END](#) }

Public Member Functions

- [PhotometricInterpretation](#) ([PIType](#) pi=[UNKNOWN](#))
- unsigned short [GetSamplesPerPixel](#) () const
return the value for Sample Per Pixel associated with a particular Photometric Interpretation
- const char * [GetString](#) () const
- [PIType](#) [GetType](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsSameColorSpace](#) ([PhotometricInterpretation](#) const &pi) const
- [operator PIType](#) () const

Static Public Member Functions

- static const char * [GetPIString](#) ([PIType](#) pi)
- static [PIType](#) [GetPIType](#) (const char *pi)
- static bool [IsRetired](#) ([PIType](#) pi)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &pi)

12.234.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractImageRegion.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [HelloVizWorld.cxx](#), [MpegVideoInfo.cs](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

12.234.2 Member Enumeration Documentation

12.234.2.1 PIType

enum [gdcmm::PhotometricInterpretation::PIType](#)

Enumerator

UNKNOWN	
MONOCHROME1	
MONOCHROME2	

PALETTE_COLOR	
RGB	
HSV	
ARGB	
CMYK	
YBR_FULL	
YBR_FULL_422	
YBR_PARTIAL_422	
YBR_PARTIAL_420	
YBR_ICT	
YBR_RCT	
PI_END	

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), and [MpegVideoInfo.cs](#).

12.234.3 Constructor & Destructor Documentation

12.234.3.1 PhotometricInterpretation()

```
gdcmm::PhotometricInterpretation::PhotometricInterpretation (
    PIType pi = UNKNOWN) [inline]
```

References [UNKNOWN](#).

Referenced by [GetSamplesPerPixel\(\)](#), [IsSameColorSpace\(\)](#), and [operator<<](#).

12.234.4 Member Function Documentation

12.234.4.1 GetPIString()

```
const char * gdcmm::PhotometricInterpretation::GetPIString (
    PIType pi) [static]
```

Referenced by [operator<<](#).

12.234.4.2 GetPIType()

```
PIType gdcmm::PhotometricInterpretation::GetPIType (
    const char * pi) [static]
```

12.234.4.3 GetSamplesPerPixel()

unsigned short gdcmm::PhotometricInterpretation::GetSamplesPerPixel () const

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

References [PhotometricInterpretation\(\)](#), and [operator<<](#).

12.234.4.4 GetString()

const char * gdcmm::PhotometricInterpretation::GetString () const

12.234.4.5 GetType()

[PIType](#) gdcmm::PhotometricInterpretation::GetType () const [inline]

12.234.4.6 IsLossless()

bool gdcmm::PhotometricInterpretation::IsLossless () const

12.234.4.7 IsLossy()

bool gdcmm::PhotometricInterpretation::IsLossy () const

12.234.4.8 IsRetired()

bool gdcmm::PhotometricInterpretation::IsRetired (
 [PIType](#) pi) [static]

12.234.4.9 IsSameColorSpace()

bool gdcmm::PhotometricInterpretation::IsSameColorSpace (
 [PhotometricInterpretation](#) const & pi) const

References [PhotometricInterpretation\(\)](#).

12.234.4.10 operator PIType()

gdcmm::PhotometricInterpretation::operator [PIType](#) () const [inline]

12.234.5 Friends And Related Symbol Documentation

12.234.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const PhotometricInterpretation & pi) [friend]
```

References [PhotometricInterpretation\(\)](#), and [GetPIString\(\)](#).

Referenced by [GetSamplesPerPixel\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmPhotometricInterpretation.h](#)

12.235 gdcm::PixelFormat Class Reference

[PixelFormat](#).

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum [ScalarType](#) {
 [UINT8](#) ,
 [INT8](#) ,
 [UINT12](#) ,
 [INT12](#) ,
 [UINT16](#) ,
 [INT16](#) ,
 [UINT32](#) ,
 [INT32](#) ,
 [UINT64](#) ,
 [INT64](#) ,
 [FLOAT16](#) ,
 [FLOAT32](#) ,
 [FLOAT64](#) ,
 [SINGLEBIT](#) ,
 [UNKNOWN](#) }

Public Member Functions

- [PixelFormat](#) ()
- [PixelFormat](#) ([ScalarType](#) st)
- [PixelFormat](#) (unsigned short samplesperpixel, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- unsigned short [GetBitsAllocated](#) () const
BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.
- unsigned short [GetBitsStored](#) () const
BitsStored see [Tag](#) (0028,0101) US Bits Stored.
- unsigned short [GetHighBit](#) () const
HighBit see [Tag](#) (0028,0102) US High Bit.
- int64_t [GetMax](#) () const
return the max possible of the pixel
- int64_t [GetMin](#) () const
return the min possible of the pixel
- unsigned short [GetPixelRepresentation](#) () const
PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.
- uint8_t [GetPixelSize](#) () const
- unsigned short [GetSamplesPerPixel](#) () const
- [ScalarType](#) [GetScalarType](#) () const
[ScalarType](#) does not take into account the sample per pixel.
- const char * [GetScalarTypeAsString](#) () const
- bool [IsCompatible](#) (const [TransferSyntax](#) &ts) const
- bool [IsValid](#) () const
return IsValid
- [operator](#) [ScalarType](#) () const
- bool [operator!=](#) (const [PixelFormat](#) &pf) const
- bool [operator!=](#) ([ScalarType](#) st) const
- bool [operator==](#) (const [PixelFormat](#) &pf) const
- bool [operator==](#) ([ScalarType](#) st) const
- void [Print](#) (std::ostream &os) const
Print.
- void [SetBitsAllocated](#) (unsigned short ba)
- void [SetBitsStored](#) (unsigned short bs)
- void [SetHighBit](#) (unsigned short hb)
- void [SetPixelRepresentation](#) (unsigned short pr)
- void [SetSamplesPerPixel](#) (unsigned short spp)
- void [SetScalarType](#) ([ScalarType](#) st)

Protected Member Functions

- bool [Validate](#) ()
When image with 24/24/23 was read, need to validate.

Friends

- class [Bitmap](#)
- std::ostream & [operator<<](#) (std::ostream &_os, const [PixelFormat](#) &pf)

12.235.1 Detailed Description

[PixelFormat](#).

By default the Pixel [Type](#) will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Fundamentally [PixelFormat](#) is very close to what DICOM allows. It will be very hard to extend this class for the upcoming DICOM standard where Floating 32 and 64bits will be allowed.

It is also very hard for this class to fully support 64bits integer type (see GetMin / GetMax signature restricted to 64bits signed).

Examples

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [DecompressImage.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FixJAIBugJPEGLS.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [MpegVideoInfo.cs](#), [RescaleImage.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcm.cxx](#).

12.235.2 Member Enumeration Documentation

12.235.2.1 ScalarType

enum [gdcm::PixelFormat::ScalarType](#)

Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	

INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

Examples

[GetArray.cs](#).

12.235.3 Constructor & Destructor Documentation

12.235.3.1 PixelFormat() [1/3]

gdcm::PixelFormat::PixelFormat () [inline]

References [PixelFormat\(\)](#).

Referenced by [PixelFormat\(\)](#), [Bitmap](#), [operator!=\(\)](#), [operator<<](#), and [operator==\(\)](#).

12.235.3.2 PixelFormat() [2/3]

```
gdcm::PixelFormat::PixelFormat (
    unsigned short samplesperpixel,
    unsigned short bitsallocated = 8,
    unsigned short bitsstored = 8,
    unsigned short highbit = 7,
    unsigned short pixelrepresentation = 0) [inline], [explicit]
```

12.235.3.3 PixelFormat() [3/3]

```
gdcm::PixelFormat::PixelFormat (
    ScalarType st)
```

12.235.4 Member Function Documentation

12.235.4.1 GetBitsAllocated()

unsigned short gdcm::PixelFormat::GetBitsAllocated () const [inline]

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

Examples

[GetJPEGSamplePrecision.cxx](#).

12.235.4.2 GetBitsStored()

unsigned short gdcm::PixelFormat::GetBitsStored () const [inline]

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

Examples

[GetJPEGSamplePrecision.cxx](#).

References [gdcm_assert](#).

12.235.4.3 GetHighBit()

unsigned short gdcm::PixelFormat::GetHighBit () const [inline]

HighBit see [Tag](#) (0028,0102) US High Bit.

References [gdcm_assert](#).

12.235.4.4 GetMax()

int64_t gdcm::PixelFormat::GetMax () const

return the max possible of the pixel

12.235.4.5 GetMin()

int64_t gdcm::PixelFormat::GetMin () const

return the min possible of the pixel

12.235.4.6 GetPixelRepresentation()

unsigned short gdcm::PixelFormat::GetPixelRepresentation () const [inline]

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

12.235.4.7 GetPixelSize()

uint8_t gdcm::PixelFormat::GetPixelSize () const

return the size of the pixel This is the number of words it would take to store one pixel

Warning

the return value takes into account the SamplesPerPixel
in the rare case when BitsAllocated == 12, the function assume word padding and value returned will be identical as if BitsAllocated == 16

Examples

[ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), and [threadgdcm.cxx](#).

12.235.4.8 GetSamplesPerPixel()

unsigned short gdcm::PixelFormat::GetSamplesPerPixel () const

Samples Per Pixel see (0028,0002) US Samples Per Pixel DICOM - only allows 1, 3 and 4 as valid value. Other value are undefined behavior.

Examples

[threadgdcm.cxx](#).

12.235.4.9 GetScalarType()

ScalarType gdcm::PixelFormat::GetScalarType () const

ScalarType does not take into account the sample per pixel.

Examples

[GetArray.cs](#).

Referenced by [operator ScalarType\(\)](#), [operator!=\(\)](#), and [operator==\(\)](#).

12.235.4.10 GetScalarTypeAsString()

const char * gdcm::PixelFormat::GetScalarTypeAsString () const

Examples

[GetArray.cs](#).

12.235.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
    const TransferSyntax & ts) const
```

12.235.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid () const

return IsValid
```

12.235.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType () const [inline]
```

References [GetScalarType\(\)](#).

12.235.4.14 operator"!=() [1/2]

```
bool gdcm::PixelFormat::operator!= (
    const PixelFormat & pf) const [inline]
```

References [PixelFormat\(\)](#).

12.235.4.15 operator"!=() [2/2]

```
bool gdcm::PixelFormat::operator!= (
    ScalarType st) const [inline]
```

References [GetScalarType\(\)](#).

12.235.4.16 operator==() [1/2]

```
bool gdcm::PixelFormat::operator== (
    const PixelFormat & pf) const [inline]
```

References [PixelFormat\(\)](#).

12.235.4.17 operator==() [2/2]

```
bool gdcm::PixelFormat::operator== (
    ScalarType st) const [inline]
```

References [GetScalarType\(\)](#).

12.235.4.18 Print()

```
void gdcm::PixelFormat::Print (  
    std::ostream & os) const
```

Print.

Referenced by [operator<<](#).

12.235.4.19 SetBitsAllocated()

```
void gdcm::PixelFormat::SetBitsAllocated (  
    unsigned short ba) [inline]
```

12.235.4.20 SetBitsStored()

```
void gdcm::PixelFormat::SetBitsStored (  
    unsigned short bs) [inline]
```

References [SetHighBit\(\)](#).

12.235.4.21 SetHighBit()

```
void gdcm::PixelFormat::SetHighBit (  
    unsigned short hb) [inline]
```

Referenced by [SetBitsStored\(\)](#).

12.235.4.22 SetPixelRepresentation()

```
void gdcm::PixelFormat::SetPixelRepresentation (  
    unsigned short pr) [inline]
```

Examples

[TemplateEmptyImage.cxx](#).

12.235.4.23 SetSamplesPerPixel()

```
void gdcm::PixelFormat::SetSamplesPerPixel (  
    unsigned short spp) [inline]
```

Examples

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

References [gdcm_assert](#), and [gdcmAssertMacro](#).

12.235.4.24 SetScalarType()

```
void gdcmm::PixelFormat::SetScalarType (
    ScalarType st)
```

Set [PixelFormat](#) based only on the [ScalarType](#)

Warning

: You need to call SetScalarType before SetSamplesPerPixel

12.235.4.25 Validate()

```
bool gdcmm::PixelFormat::Validate () [protected]
```

When image with 24/24/23 was read, need to validate.

12.235.5 Friends And Related Symbol Documentation

12.235.5.1 Bitmap

```
friend class Bitmap [friend]
```

References [PixelFormat\(\)](#), [Bitmap](#), and [operator<<](#).

Referenced by [Bitmap](#).

12.235.5.2 operator<<

```
std::ostream & operator<< (
    std::ostream & __os,
    const PixelFormat & pf) [friend]
```

References [PixelFormat\(\)](#), and [Print\(\)](#).

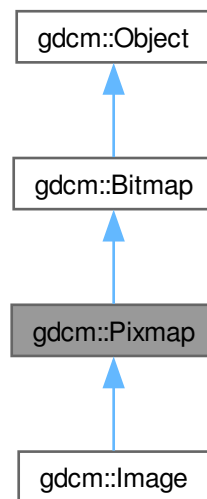
Referenced by [Bitmap](#).

The documentation for this class was generated from the following file:

- [gdcmmPixelFormat.h](#)

Pixmap class.

Inheritance diagram for `gdcm::Pixmap`:

[illegible]

Public Member Functions

- [Pixmap](#) ()
- [~Pixmap](#) () override
- bool [AreOverlaysInPixelData](#) () const override
returns if Overlays are stored in the unused bit of the pixel data:
- [Curve](#) & [GetCurve](#) (size_t i=0)
[Curve](#): group 50xx.
- const [Curve](#) & [GetCurve](#) (size_t i=0) const
- [IconImage](#) & [GetIconImage](#) ()
- const [IconImage](#) & [GetIconImage](#) () const
Set/Get Icon Image.
- size_t [GetNumberOfCurves](#) () const
- size_t [GetNumberOfOverlays](#) () const
- [Overlay](#) & [GetOverlay](#) (size_t i=0)
[Overlay](#): group 60xx.
- const [Overlay](#) & [GetOverlay](#) (size_t i=0) const
- void [Print](#) (std::ostream &) const override
- void [RemoveOverlay](#) (size_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size_t n)
- void [SetNumberOfOverlays](#) (size_t n)
- bool [UnusedBitsPresentInPixelData](#) () const override
returns if there are unused bits in the pixel data

Public Member Functions inherited from [gdcm::Bitmap](#)

- [Bitmap](#) ()
- [~Bitmap](#) () override
- void [Clear](#) ()
- bool [GetBuffer](#) (char *buffer) const
Access the raw data.
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- const [DataElement](#) & [GetDataElement](#) () const
- unsigned int [GetDimension](#) (unsigned int idx) const
- const unsigned int * [GetDimensions](#) () const
Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- [LookupTable](#) & [GetLUT](#) ()
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
INTERNAL do not use.
- unsigned int [GetNumberOfDimensions](#) () const
Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
return the photometric interpretation
- [PixelFormat](#) & [GetPixelFormat](#) ()

- const [PixelFormat](#) & [GetPixelFormat](#) () const
Get/Set [PixelFormat](#).
- unsigned int [GetPlanarConfiguration](#) () const
return the planar configuration
- unsigned int [GetRows](#) () const
- const [TransferSyntax](#) & [GetTransferSyntax](#) () const
- bool [IsEmpty](#) () const
- bool [IsLossy](#) () const
Return whether or not the image was compressed using a lossy compressor or not.
- bool [IsTransferSyntaxCompatible](#) ([TransferSyntax](#) const &ts) const
- void [SetColumns](#) (unsigned int col)
- void [SetDataElement](#) ([DataElement](#) const &de)
- void [SetDimension](#) (unsigned int idx, unsigned int dim)
- void [SetDimensions](#) (const unsigned int dims[3])
- void [SetLossyFlag](#) (bool f)
Specifically set that the image was compressed using a lossy compression mechanism.
- void [SetLUT](#) ([LookupTable](#) const &lut)
Set/Get LUT.
- void [SetNeedByteSwap](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)
- void [SetRows](#) (unsigned int rows)
- void [SetTransferSyntax](#) ([TransferSyntax](#) const &ts)
Transfer syntax.

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Protected Attributes

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)

Protected Attributes inherited from [gdcm::Bitmap](#)

- `std::vector< unsigned int >` [Dimensions](#)
- `bool` [LossyFlag](#)
- `LUTPtr` [LUT](#)
- `bool` [NeedByteSwap](#)
- `unsigned int` [NumberOfDimensions](#)
- `PixelFormat` [PF](#)
- `PhotometricInterpretation` [PI](#)
- `DataElement` [PixelData](#)
- `unsigned int` [PlanarConfiguration](#)
- `TransferSyntax` [TS](#)

Additional Inherited Members

Protected Types inherited from [gdcm::Bitmap](#)

- `typedef` [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions inherited from [gdcm::Bitmap](#)

- `bool` [ComputeLossyFlag](#) ()
- `bool` [GetBuffer2](#) (std::ostream &os) const
- `bool` [TryJPEG2000Codec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryJPEG2000Codec2](#) (std::ostream &os) const
- `bool` [TryJPEGCodec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryJPEGCodec2](#) (std::ostream &os) const
- `bool` [TryJPEGLSCCodec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryKAKADUCoec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryPVRGCodec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryRAWCodec](#) (char *buffer, bool &lossyflag) const
- `bool` [TryRLECodec](#) (char *buffer, bool &lossyflag) const

Protected Member Functions inherited from [gdcm::Object](#)

- `void` [Register](#) ()
- `void` [UnRegister](#) ()

12.236.1 Detailed Description

[Pixmap](#) class.

A bitmap based image. Used as parent for both [IconImage](#) and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

See also

[PixmapReader](#)

Examples

[FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), and [StandardizeFiles.cs](#).

12.236.2 Constructor & Destructor Documentation

12.236.2.1 Pixmap()

gdcm::Pixmap::Pixmap ()

12.236.2.2 ~Pixmap()

gdcm::Pixmap::~Pixmap () [override]

12.236.3 Member Function Documentation

12.236.3.1 AreOverlaysInPixelData()

bool gdcm::Pixmap::AreOverlaysInPixelData () const [override], [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

12.236.3.2 GetCurve() [1/2]

[Curve](#) & gdcm::Pixmap::GetCurve (
size_t i = 0) [inline]

[Curve](#): group 50xx.

References [Curves](#), and [gdcm_assert](#).

12.236.3.3 GetCurve() [2/2]

const [Curve](#) & gdcm::Pixmap::GetCurve (
size_t i = 0) const [inline]

References [Curves](#), and [gdcm_assert](#).

12.236.3.4 GetIconImage() [1/2]

[IconImage](#) & gdcm::Pixmap::GetIconImage () [inline]

References [Icon](#).

12.236.3.5 GetIconImage() [2/2]

```
const IconImage & gdcm::Pixmap::GetIconImage () const [inline]
```

Set/Get Icon [Image](#).

References [Icon](#).

12.236.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves () const [inline]
```

References [Curves](#).

12.236.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays () const [inline]
```

References [Overlays](#).

12.236.3.8 GetOverlay() [1/2]

```
Overlay & gdcm::Pixmap::GetOverlay (  
    size_t i = 0) [inline]
```

[Overlay](#): group 60xx.

References [gdcm_assert](#), and [Overlays](#).

12.236.3.9 GetOverlay() [2/2]

```
const Overlay & gdcm::Pixmap::GetOverlay (  
    size_t i = 0) const [inline]
```

References [gdcm_assert](#), and [Overlays](#).

12.236.3.10 Print()

```
void gdcm::Pixmap::Print (  
    std::ostream & ) const [override], [virtual]
```

Reimplemented from [gdcm::Bitmap](#).

12.236.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (  
    size_t i) [inline]
```

References [gdcm_assert](#), and [Overlays](#).

12.236.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (  
    IconImage const & ii) [inline]
```

References [Icon](#).

12.236.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (  
    size_t n) [inline]
```

References [Curves](#).

12.236.3.14 SetNumberOfOverlays()

```
void gdcm::Pixmap::SetNumberOfOverlays (  
    size_t n) [inline]
```

References [Overlays](#).

12.236.3.15 UnusedBitsPresentInPixelData()

```
bool gdcm::Pixmap::UnusedBitsPresentInPixelData () const [override], [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from [gdcm::Bitmap](#).

12.236.4 Member Data Documentation

12.236.4.1 Curves

```
std::vector<Curve> gdcm::Pixmap::Curves [protected]
```

Referenced by [GetCurve\(\)](#), [GetCurve\(\)](#), [GetNumberOfCurves\(\)](#), and [SetNumberOfCurves\(\)](#).

12.236.4.2 Icon

[SmartPointer<IconImage>](#) gdcmm::Pixmap::Icon [protected]

Referenced by [GetIconImage\(\)](#), [GetIconImage\(\)](#), and [SetIconImage\(\)](#).

12.236.4.3 Overlays

`std::vector<Overlay>` gdcmm::Pixmap::Overlays [protected]

Referenced by [GetNumberOfOverlays\(\)](#), [GetOverlay\(\)](#), [GetOverlay\(\)](#), [RemoveOverlay\(\)](#), and [SetNumberOfOverlays\(\)](#).

The documentation for this class was generated from the following file:

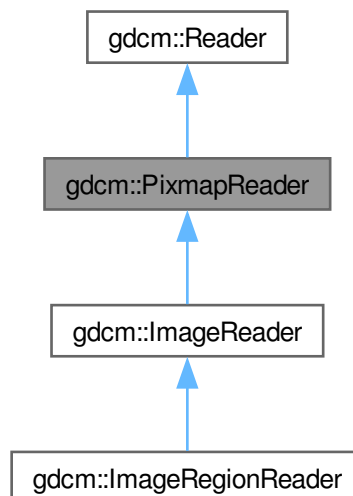
- [gdcmmPixmap.h](#)

12.237 gdcmm::PixmapReader Class Reference

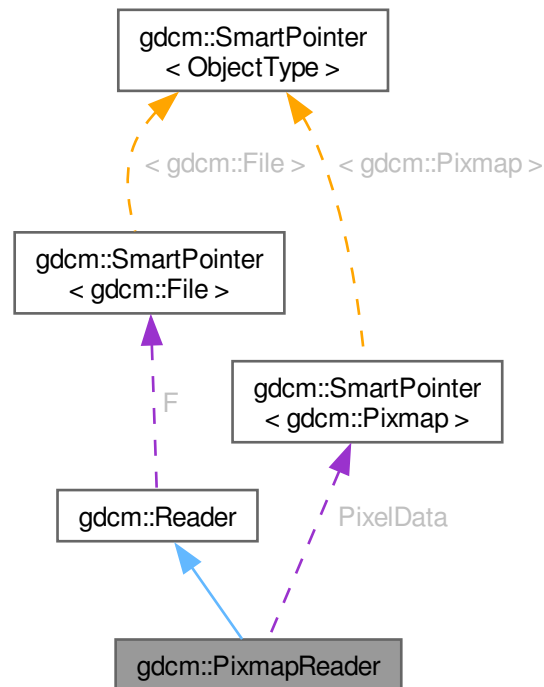
[PixmapReader](#).

```
#include <gdcmmPixmapReader.h>
```

Inheritance diagram for gdcmm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- [PixmapReader](#) ()
- [~PixmapReader](#) () override
- [Pixmap](#) & [GetPixmap](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
Return the read image (need to call [Read\(\)](#) first).
- bool [Read](#) () override

Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()
Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
Set/Get [File](#).

- `size_t GetStreamCurrentPosition () const`
- `bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)`
Will only read the specified selected private tags.
- `bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)`
Will only read the specified selected tags.
- `bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())`
- `void SetFile (File &file)`
Set/Get File.
- `void SetFileName (const char *filename__native)`
- `void SetStream (std::istream &input__stream)`
Set the open-ed stream directly.

Protected Member Functions

- `virtual bool ReadACRNEMAIImage ()`
- `virtual bool ReadImage (MediaStorage const &ms)`
- `bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)`

Protected Member Functions inherited from [gdcm::Reader](#)

- `std::istream * GetStreamPtr () const`
- `bool ReadDataSet ()`
- `bool ReadMetaInformation ()`
- `bool ReadPreamble ()`

Protected Attributes

- `SmartPointer< Pixmap > PixelData`

Protected Attributes inherited from [gdcm::Reader](#)

- `SmartPointer< File > F`

12.237.1 Detailed Description

[PixmapReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [Pixmap](#) representation By default it is also loading the lookup table and overlay when found as they impact the rendering or the image

See PS 3.3-2008, [Table C.7-11b](#) IMAGE PIXEL MACRO ATTRIBUTES for the list of attribute that belong to what gdcm calls a '[Pixmap](#)'

Warning

the API `ReadUpToTag` and `ReadSelectedTag`

See also

[Pixmap](#)

Examples

[StandardizeFiles.cs](#).

12.237.2 Constructor & Destructor Documentation

12.237.2.1 PixmapReader()

gdcm::PixmapReader::PixmapReader ()

12.237.2.2 ~PixmapReader()

gdcm::PixmapReader::~~PixmapReader () [override]

12.237.3 Member Function Documentation

12.237.3.1 GetPixmap() [1/2]

[Pixmap](#) & gdcm::PixmapReader::GetPixmap ()

12.237.3.2 GetPixmap() [2/2]

const [Pixmap](#) & gdcm::PixmapReader::GetPixmap () const

Return the read image (need to call [Read\(\)](#) first).

Examples

[StandardizeFiles.cs](#).

12.237.3.3 Read()

bool gdcm::PixmapReader::Read () [override], [virtual]

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an [Pixmap](#).

Reimplemented from [gdcm::Reader](#).

Examples

[StandardizeFiles.cs](#).

12.237.3.4 ReadACRNEMAIImage()

virtual bool gdcm::PixmapReader::ReadACRNEMAIImage () [protected], [virtual]

Reimplemented in [gdcm::ImageReader](#).

12.237.3.5 ReadImage()

virtual bool gdcm::PixmapReader::ReadImage (
[MediaStorage](#) const & ms) [protected], [virtual]

Reimplemented in [gdcm::ImageReader](#).

12.237.3.6 ReadImageInternal()

bool gdcm::PixmapReader::ReadImageInternal (
[MediaStorage](#) const & ms,
 bool handlepixeldata = true) [protected]

12.237.4 Member Data Documentation

12.237.4.1 PixelData

[SmartPointer<Pixmap>](#) gdcm::PixmapReader::PixelData [protected]

The documentation for this class was generated from the following file:

- [gdcmPixmapReader.h](#)

12.238 gdcm::PixmapToPixmapFilter Class Reference

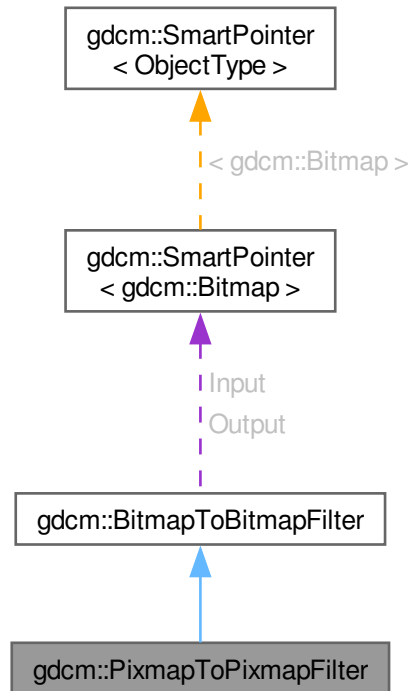
[PixmapToPixmapFilter](#) class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- [PixmapToPixmapFilter \(\)](#)
- [~PixmapToPixmapFilter \(\)](#)=default
- [Pixmap & GetInput \(\)](#)
- [const Pixmap & GetOutput \(\)](#) const
Get Output image.
- [const Pixmap & GetOutputAsPixmap \(\)](#) const

Public Member Functions inherited from [gdcm::BitmapToBitmapFilter](#)

- [BitmapToBitmapFilter \(\)](#)
- [~BitmapToBitmapFilter \(\)](#)=default
- [const Bitmap & GetOutput \(\)](#) const
Get Output image.
- [const Bitmap & GetOutputAsBitmap \(\)](#) const
- [void SetInput \(const Bitmap &image\)](#)
Set input image.

Additional Inherited Members

Protected Attributes inherited from [gdcm::BitmapToBitmapFilter](#)

- [SmartPointer< Bitmap > Input](#)
- [SmartPointer< Bitmap > Output](#)

12.238.1 Detailed Description

[PixmapToPixmapFilter](#) class.

Super class for all filter taking an image and producing an output image

Examples

[StandardizeFiles.cs](#).

12.238.2 Constructor & Destructor Documentation

12.238.2.1 PixmapToPixmapFilter()

`gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ()`

12.238.2.2 ~PixmapToPixmapFilter()

`gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter () [default]`

12.238.3 Member Function Documentation

12.238.3.1 GetInput()

[Pixmap](#) & `gdcm::PixmapToPixmapFilter::GetInput ()`

12.238.3.2 GetOutput()

`const Pixmap & gdcm::PixmapToPixmapFilter::GetOutput () const`

Get Output image.

12.238.3.3 GetOutputAsPixmap()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutputAsPixmap () const
```

The documentation for this class was generated from the following file:

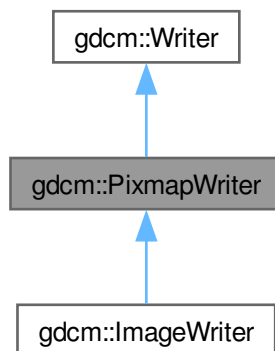
- [gdcmPixmapToPixmapFilter.h](#)

12.239 gdcm::PixmapWriter Class Reference

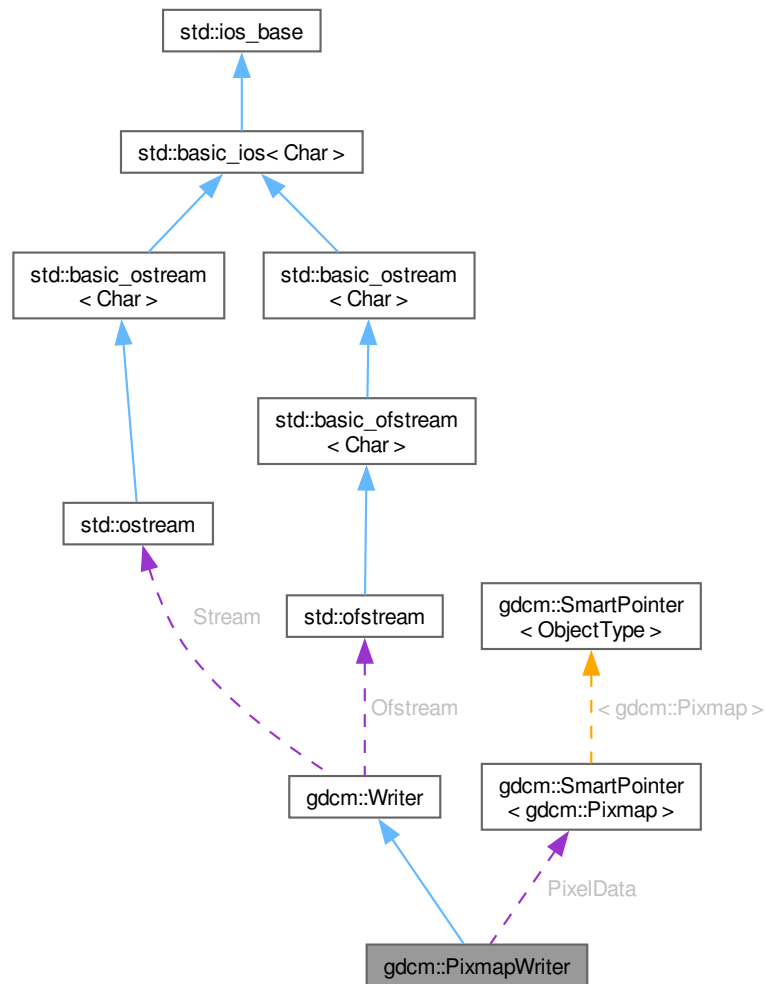
[PixmapWriter](#).

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for `gdcm::PixmapWriter`:



Public Member Functions

- `PixmapWriter ()`
- `~PixmapWriter ()` override
- virtual `Pixmap & GetImage ()`
- virtual const `Pixmap & GetImage ()` const
- `Pixmap & GetPixmap ()`
- const `Pixmap & GetPixmap ()` const
- virtual void `SetImage (Pixmap const &img)`
- void `SetPixmap (Pixmap const &img)`
- bool `Write ()` override

Write.

Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)
Undocumented function, do not use (= leave default).
- void [SetFile](#) (const [File](#) &f)
Set/Get the DICOM file ([DataSet](#) + Header).
- void [SetFileName](#) (const char *filename__native)
Set the filename of DICOM file to write:
- void [SetStream](#) (std::ostream &output_stream)
Set user ostream buffer.

Protected Member Functions

- void [DoIconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

Protected Attributes inherited from [gdcm::Writer](#)

- std::ofstream * [Ofstream](#)
- std::ostream * [Stream](#)

12.239.1 Detailed Description

[PixmapWriter](#).

This class will takes two inputs:

1. The DICOM [DataSet](#)
2. The [Image](#) input It will override any info from the [Image](#) over the [DataSet](#).

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

Examples

[StandardizeFiles.cs](#).

12.239.2 Constructor & Destructor Documentation

12.239.2.1 PixmapWriter()

`gdcm::PixmapWriter::PixmapWriter ()`

12.239.2.2 ~PixmapWriter()

`gdcm::PixmapWriter::~PixmapWriter ()` [override]

12.239.3 Member Function Documentation

12.239.3.1 DoIconImage()

`void gdcm::PixmapWriter::DoIconImage (`
 [DataSet](#) & ds,
 [Pixmap](#) const & image) [protected]

12.239.3.2 GetImage() [1/2]

`virtual Pixmap & gdcm::PixmapWriter::GetImage ()` [inline], [virtual]

Reimplemented in [gdcm::ImageWriter](#).

References [PixelData](#).

12.239.3.3 GetImage() [2/2]

`virtual const Pixmap & gdcm::PixmapWriter::GetImage () const` [inline], [virtual]

Set/Get [Pixmap](#) to be written It will overwrite anything [Pixmap](#) infos found in [DataSet](#) (see parent class to see how to pass dataset)

Reimplemented in [gdcm::ImageWriter](#).

References [PixelData](#).

12.239.3.4 GetPixmap() [1/2]

`Pixmap & gdcm::PixmapWriter::GetPixmap ()` [inline]

References [PixelData](#).

12.239.3.5 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapWriter::GetPixmap () const    [inline]
```

References [PixelData](#).

12.239.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (  
    MediaStorage const & refms)    [protected]
```

12.239.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (  
    Pixmap const & img)    [virtual]
```

Examples

[BasicImageAnonymizer.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [DecompressImage.cs](#),
[GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#),
and [TemplateEmptyImage.cxx](#).

12.239.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (  
    Pixmap const & img)
```

Examples

[StandardizeFiles.cs](#).

12.239.3.9 Write()

```
bool gdcm::PixmapWriter::Write ()    [override], [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

Examples

[StandardizeFiles.cs](#).

12.239.4 Member Data Documentation

12.239.4.1 PixelData

[SmartPointer<Pixmap>](#) `gdcm::PixmapWriter::PixelData` [protected]

Referenced by [GetImage\(\)](#), [GetImage\(\)](#), [GetPixmap\(\)](#), and [GetPixmap\(\)](#).

The documentation for this class was generated from the following file:

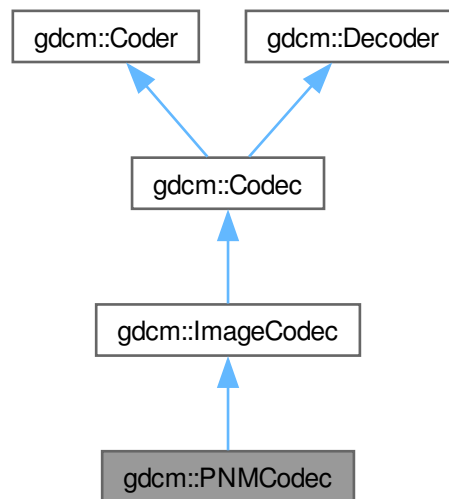
- [gdcmPixmapWriter.h](#)

12.240 gdcm::PNMCodec Class Reference

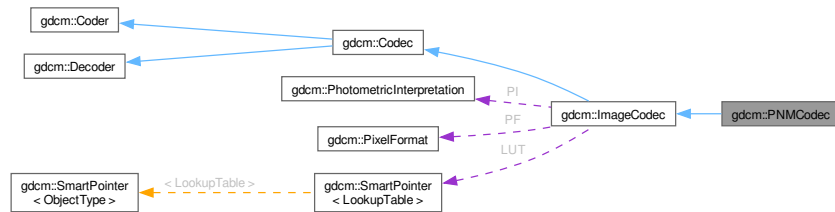
Class to do PNM.

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for `gdcm::PNMCodec`:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- [PNMCodec](#) ()
- [~PNMCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- bool [Read](#) (const char *filename, [DataElement](#) &out) const
- void [SetBufferLength](#) (unsigned long l)
- bool [Write](#) (const char *filename, const [DataElement](#) &out) const

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- bool [Decode](#) ([DataElement](#) const &is_, [DataElement](#) &os) override
Decode.
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)

- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default
- virtual bool [Code](#) ([DataElement](#) const &in_, [DataElement](#) &out_)
Code.

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.240.1 Detailed Description

Class to do PNM.

PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.↵

Note

Only support P5 & P6 PNM file (binary grayscale and binary rgb)

Examples

[ExtractIconFromFile.cxx](#).

12.240.2 Constructor & Destructor Documentation

12.240.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ()
```

12.240.2.2 ~PNMCodec()

```
gdcm::PNMCodec::~~PNMCodec () [override]
```

12.240.3 Member Function Documentation

12.240.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.240.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
    TransferSyntax const & ) const    [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.240.3.3 Clone()

```
ImageCodec * gdcm::PNMCodec::Clone () const    [override], [virtual]
```

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.240.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength () const    [inline]
```

12.240.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts)    [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.240.3.6 Read()

```
bool gdcm::PNMCodec::Read (
    const char * filename,
    DataElement & out) const
```

12.240.3.7 SetBufferLength()

```
void gdcm::PNMCodec::SetBufferLength (
    unsigned long l)    [inline]
```


12.240.3.8 Write()

```
bool gdcm::PNMCodec::Write (
    const char * filename,
    const DataElement & out) const
```

Examples

[ExtractIconFromFile.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmPNMCodec.h](#)

12.241 gdcm::Preamble Class Reference

DICOM [Preamble](#) (Part 10).

```
#include <gdcmPreamble.h>
```

Public Member Functions

- [Preamble](#) ()
- [Preamble](#) (Preamble const &)
- [~Preamble](#) ()
- void [Clear](#) ()
Clear.
- void [Create](#) ()
- const char * [GetInternal](#) () const
Get internal pointer to preamble.
- [VL GetLength](#) () const
Return size of [Preamble](#).
- bool [IsEmpty](#) () const
Check if [Preamble](#) is empty.
- [Preamble](#) & [operator=](#) ([Preamble](#) const &)
- void [Print](#) (std::ostream &os) const
Print [Preamble](#).
- std::istream & [Read](#) (std::istream &is)
Read [Preamble](#).
- void [Remove](#) ()
- void [Valid](#) ()
Set [Preamble](#) to the default one.
- std::ostream const & [Write](#) (std::ostream &os) const
Write [Preamble](#).

Protected Member Functions

- bool [IsValid](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Preamble](#) &_val)

12.241.1 Detailed Description

DICOM [Preamble](#) (Part 10).

12.241.2 Constructor & Destructor Documentation

12.241.2.1 Preamble() [1/2]

gdcm::Preamble::Preamble ()

Referenced by [Preamble\(\)](#), [~Preamble\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.241.2.2 ~Preamble()

gdcm::Preamble::~~Preamble ()

References [Preamble\(\)](#), and [operator<<](#).

12.241.2.3 Preamble() [2/2]

gdcm::Preamble::Preamble (
Preamble const &) [inline]

References [Preamble\(\)](#), and [Create\(\)](#).

12.241.3 Member Function Documentation

12.241.3.1 Clear()

void gdcm::Preamble::Clear ()

Clear.

12.241.3.2 Create()

```
void gdcmm::Preamble::Create ()
```

Referenced by [Preamble\(\)](#), and [operator=\(\)](#).

12.241.3.3 GetInternal()

```
const char * gdcmm::Preamble::GetInternal () const [inline]
```

Get internal pointer to preamble.

12.241.3.4 GetLength()

```
VL gdcmm::Preamble::GetLength () const [inline]
```

Return size of [Preamble](#).

12.241.3.5 IsEmpty()

```
bool gdcmm::Preamble::IsEmpty () const [inline]
```

Check if [Preamble](#) is empty.

12.241.3.6 IsValid()

```
bool gdcmm::Preamble::IsValid () const [inline], [protected]
```

12.241.3.7 operator=()

```
Preamble & gdcmm::Preamble::operator= (  
    Preamble const & ) [inline]
```

References [Preamble\(\)](#), and [Create\(\)](#).

12.241.3.8 Print()

```
void gdcmm::Preamble::Print (  
    std::ostream & os) const
```

Print [Preamble](#).

12.241.3.9 Read()

```
std::istream & gdcmm::Preamble::Read (  
    std::istream & is)
```

Read [Preamble](#).

12.241.3.10 Remove()

```
void gdcmm::Preamble::Remove ()
```

12.241.3.11 Valid()

```
void gdcmm::Preamble::Valid ()
```

Set [Preamble](#) to the default one.

12.241.3.12 Write()

```
std::ostream const & gdcmm::Preamble::Write (  
    std::ostream & os) const
```

Write [Preamble](#).

12.241.4 Friends And Related Symbol Documentation

12.241.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Preamble & _val) [friend]
```

References [Preamble\(\)](#).

Referenced by [~Preamble\(\)](#).

The documentation for this class was generated from the following file:

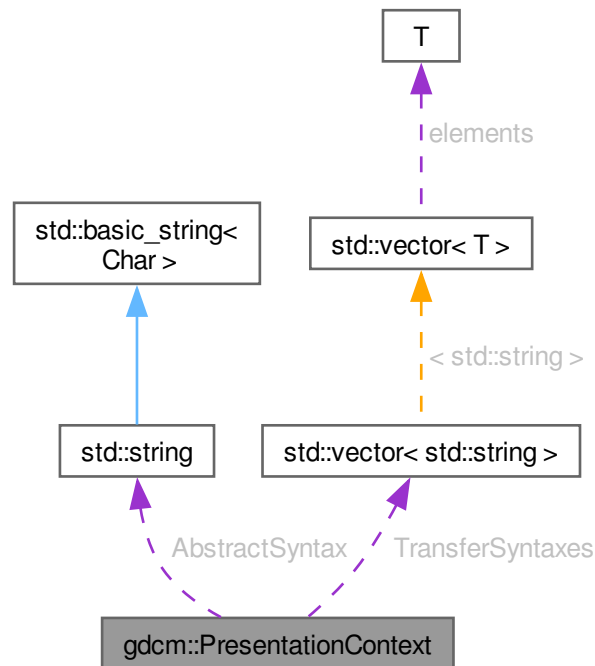
- [gdcmmPreamble.h](#)

12.242 gdcm::PresentationContext Class Reference

[PresentationContext](#).

```
#include <gdcmPresentationContext.h>
```

Collaboration diagram for gdcm::PresentationContext:



Public Types

- typedef TransferSyntaxArrayType::size_type [SizeType](#)
- typedef std::vector< std::string > [TransferSyntaxArrayType](#)

Public Member Functions

- [PresentationContext](#) ()
- [PresentationContext](#) (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntax)
- void [AddTransferSyntax](#) (const char *tsstr)
- const char * [GetAbstractSyntax](#) () const
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8_t [GetPresentationContextID](#) () const
- const char * [GetTransferSyntax](#) ([SizeType](#) i) const
- bool [operator==](#) (const [PresentationContext](#) &pc) const
- void [Print](#) (std::ostream &os) const
- void [SetAbstractSyntax](#) (const char *absyn)
- void [SetPresentationContextID](#) (uint8_t id)

Protected Attributes

- `std::string` [AbstractSyntax](#)
- `uint8_t` [ID](#)
- `std::vector< std::string >` [TransferSyntaxes](#)

12.242.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

12.242.2 Member Typedef Documentation

12.242.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

12.242.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

12.242.3 Constructor & Destructor Documentation

12.242.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ()
```

Referenced by [operator==\(\)](#).

12.242.3.2 PresentationContext() [2/2]

```
gdcm::PresentationContext::PresentationContext (
    UIDs::TSName asname,
    UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single [TransferSyntax](#) set to tsname (default to Implicit [VR](#) LittleEndian when not specified).

References [gdcm::UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#).

12.242.4 Member Function Documentation

12.242.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
    const char * tsstr)
```

12.242.4.2 GetAbstractSyntax()

```
const char * gdcm::PresentationContext::GetAbstractSyntax () const [inline]
```

References [AbstractSyntax](#).

12.242.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes () const [inline]
```

References [TransferSyntaxes](#).

12.242.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID () const
```

12.242.4.5 GetTransferSyntax()

```
const char * gdcm::PresentationContext::GetTransferSyntax (
    SizeType i) const [inline]
```

References [TransferSyntaxes](#).

12.242.4.6 operator==()

```
bool gdcm::PresentationContext::operator== (
    const PresentationContext & pc) const [inline]
```

References [PresentationContext\(\)](#), [AbstractSyntax](#), [gdcm_assert](#), and [TransferSyntaxes](#).

12.242.4.7 Print()

```
void gdcm::PresentationContext::Print (
    std::ostream & os) const
```

12.242.4.8 SetAbstractSyntax()

```
void gdcmm::PresentationContext::SetAbstractSyntax (  
    const char * absyn) [inline]
```

References [AbstractSyntax](#).

12.242.4.9 SetPresentationContextID()

```
void gdcmm::PresentationContext::SetPresentationContextID (  
    uint8_t id)
```

12.242.5 Member Data Documentation

12.242.5.1 AbstractSyntax

```
std::string gdcmm::PresentationContext::AbstractSyntax [protected]
```

Referenced by [GetAbstractSyntax\(\)](#), [operator==\(\)](#), and [SetAbstractSyntax\(\)](#).

12.242.5.2 ID

```
uint8_t gdcmm::PresentationContext::ID [protected]
```

12.242.5.3 TransferSyntaxes

```
std::vector<std::string> gdcmm::PresentationContext::TransferSyntaxes [protected]
```

Referenced by [GetNumberOfTransferSyntaxes\(\)](#), [GetTransferSyntax\(\)](#), and [operator==\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmPresentationContext.h](#)

12.243 gdcmm::network::PresentationContextAC Class Reference

[PresentationContextAC](#).

```
#include <gdcmmPresentationContextAC.h>
```


Public Member Functions

- [PresentationContextAC](#) ()
- uint8_t [GetPresentationContextID](#) () const
- uint8_t [GetReason](#) () const
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetPresentationContextID](#) (uint8_t id)
- void [SetReason](#) (uint8_t r)
- void [SetTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.243.1 Detailed Description

[PresentationContextAC](#).

[Table 9-18](#) PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContext](#)

12.243.2 Constructor & Destructor Documentation

12.243.2.1 PresentationContextAC()

`gdcmm::network::PresentationContextAC::PresentationContextAC ()`

12.243.3 Member Function Documentation

12.243.3.1 GetPresentationContextID()

`uint8_t gdcmm::network::PresentationContextAC::GetPresentationContextID () const` [\[inline\]](#)

12.243.3.2 GetReason()

`uint8_t gdcmm::network::PresentationContextAC::GetReason () const` [\[inline\]](#)

12.243.3.3 GetTransferSyntax()

`TransferSyntaxSub const & gdcmm::network::PresentationContextAC::GetTransferSyntax () const` [\[inline\]](#)

12.243.3.4 Print()

```
void gdcm::network::PresentationContextAC::Print (  
    std::ostream & os) const
```

12.243.3.5 Read()

```
std::istream & gdcm::network::PresentationContextAC::Read (  
    std::istream & is)
```

12.243.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (  
    uint8_t id)
```

12.243.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (  
    uint8_t r) [inline]
```

12.243.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (  
    TransferSyntaxSub const & ts)
```

12.243.3.9 Size()

```
size_t gdcm::network::PresentationContextAC::Size () const
```

12.243.3.10 Write()

```
const std::ostream & gdcm::network::PresentationContextAC::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmPresentationContextAC.h](#)

12.244 gdcm::PresentationContextGenerator Class Reference

[PresentationContextGenerator](#).

```
#include <gdcmPresentationContextGenerator.h>
```

Public Types

- typedef std::vector< [PresentationContext](#) > [PresentationContextArrayType](#)
- typedef [PresentationContextArrayType::size_type](#) [SizeType](#)

Public Member Functions

- [PresentationContextGenerator](#) ()
- bool [AddFromFile](#) (const [File](#) &file)
- bool [GenerateFromFilenames](#) (const [Directory::FilenamesType](#) &files)
- bool [GenerateFromUID](#) ([UIDs::TSName](#) asname)
Generate the [PresentationContext](#) array from a UID (eg. [VerificationSOPClass](#)).
- [PresentationContextArrayType](#) const & [GetPresentationContexts](#) ()
- void [SetDefaultTransferSyntax](#) (const [TransferSyntax](#) &ts)
Not implemented for now. GDCM internally uses Implicit Little Endian.
- void [SetMergeModeToAbstractSyntax](#) ()
- void [SetMergeModeToTransferSyntax](#) ()

Protected Member Functions

- bool [AddPresentationContext](#) (const char *absyn, const char *ts)
- const char * [GetDefaultTransferSyntax](#) () const

12.244.1 Detailed Description

[PresentationContextGenerator](#).

This class is responsible for generating the proper [PresentationContext](#) that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

For example a [PresentationContext](#) will express that negotiation requires that CT [Image](#) Storage are send using JPEG Lossless, while US [Image](#) Storage are sent using RLE Transfer Syntax.

Two very different API are exposed one which will always default to little endian transfer syntax see [GenerateFromUID\(\)](#) This API is used for C-ECHO, C-FIND and C-MOVE (SCU). Another API:↵[GenerateFromFilenames\(\)](#) is used for C-STORE (SCU) as it will loop over all filenames argument to detect the actual encoding. and therefore find the proper encoding to be used.

Two modes are available. The default mode ([SetMergeModeToAbstractSyntax](#)) append [PresentationContext](#) (one [AbstractSyntax](#) and one [TransferSyntax](#)), as long a they are different. Eg MR [Image](#) Storage/JPEG2000 and MR [Image](#) Storage/JPEGLossless would be considered different. the other mode [SetMergeModeTo↵TransferSyntax](#) merge any new [TransferSyntax](#) to the already existing [PresentationContext](#) in order to re-use the same [AbstractSyntax](#).

See also

[PresentationContext](#)

Examples

[CStoreQtProgress.cxx](#).

12.244.2 Member Typedef Documentation

12.244.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType
```

12.244.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

12.244.3 Constructor & Destructor Documentation

12.244.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ()
```

12.244.4 Member Function Documentation

12.244.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
    const File & file)
```

Add a single [PresentationContext](#) from a single [File](#). Call multiple times when dealing with multiple files.

12.244.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
    const char * absyn,
    const char * ts) [protected]
```

12.244.4.3 GenerateFromFilenames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFilenames (
    const Directory::FilenamesType & files)
```

Generate the [PresentationContext](#) array from a File-Set. [File](#) specified needs to be valid DICOM files. Used for C-STORE operations

Examples

[CStoreQtProgress.cxx](#).

12.244.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
    UIDs::TSName asname)
```

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass).

12.244.4.5 GetDefaultTransferSyntax()

```
const char * gdcm::PresentationContextGenerator::GetDefaultTransferSyntax () const [protected]
```

12.244.4.6 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::PresentationContextGenerator::GetPresentationContexts () [inline]
```

Examples

[CStoreQtProgress.cxx](#).

12.244.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
    const TransferSyntax & ts)
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

12.244.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ()
```

12.244.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ()
```

The documentation for this class was generated from the following file:

- [gdcmPresentationContextGenerator.h](#)

12.245 gdcm::network::PresentationContextRQ Class Reference

[PresentationContextRQ](#).

```
#include <gdcmPresentationContextRQ.h>
```

Public Types

- typedef std::vector< [TransferSyntaxSub](#) >::size_type [SizeType](#)

Public Member Functions

- [PresentationContextRQ](#) ()
- [PresentationContextRQ](#) (const [PresentationContext](#) &pc)
- [PresentationContextRQ](#) ([UIDs::TSName](#) asname, [UIDs::TSName](#) tsname=[UIDs::ImplicitVRLittleEndianDefaultTransferSyntax](#))
- void [AddTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- [AbstractSyntax](#) & [GetAbstractSyntax](#) ()
- [AbstractSyntax](#) const & [GetAbstractSyntax](#) () const
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8_t [GetPresentationContextID](#) () const
- [TransferSyntaxSub](#) & [GetTransferSyntax](#) ([SizeType](#) i)
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) ([SizeType](#) i) const
- std::vector< [TransferSyntaxSub](#) > const & [GetTransferSyntaxes](#) () const
- bool [operator==](#) (const [PresentationContextRQ](#) &pc) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetAbstractSyntax](#) ([AbstractSyntax](#) const &absyn)
- void [SetPresentationContextID](#) (uint8_t id)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.245.1 Detailed Description

[PresentationContextRQ](#).

[Table 9-13](#) PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContextAC](#)

12.245.2 Member Typedef Documentation

12.245.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

12.245.3 Constructor & Destructor Documentation

12.245.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ()
```

Referenced by [operator==\(\)](#).

12.245.3.2 PresentationContextRQ() [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    UIDs::TSName asname,
    UIDs::TSName tname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
```

Initialize Presentation Context with [AbstractSyntax](#) set to asname and with a single [TransferSyntax](#) set to tname (default to Implicit [VR](#) LittleEndian when not specified).

References [gdcm::UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#).

12.245.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    const PresentationContext & pc)
```

12.245.4 Member Function Documentation

12.245.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
    TransferSyntaxSub const & ts)
```

12.245.4.2 GetAbstractSyntax() [1/2]

[AbstractSyntax](#) & gdcm::network::PresentationContextRQ::GetAbstractSyntax () [inline]

12.245.4.3 GetAbstractSyntax() [2/2]

[AbstractSyntax](#) const & gdcm::network::PresentationContextRQ::GetAbstractSyntax () const [inline]

12.245.4.4 GetNumberOfTransferSyntaxes()

[SizeType](#) gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes () const [inline]

12.245.4.5 GetPresentationContextID()

uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID () const

12.245.4.6 GetTransferSyntax() [1/2]

[TransferSyntaxSub](#) & gdcm::network::PresentationContextRQ::GetTransferSyntax (
 [SizeType](#) i) [inline]

12.245.4.7 GetTransferSyntax() [2/2]

```
TransferSyntaxSub const & gdcmm::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i) const    [inline]
```

12.245.4.8 GetTransferSyntaxes()

```
std::vector< TransferSyntaxSub > const & gdcmm::network::PresentationContextRQ::GetTransferSyntaxes () const    [inline]
```

12.245.4.9 operator==()

```
bool gdcmm::network::PresentationContextRQ::operator==(
    const PresentationContextRQ & pc) const    [inline]
```

References [PresentationContextRQ\(\)](#), and [gdcmm_assert](#).

12.245.4.10 Print()

```
void gdcmm::network::PresentationContextRQ::Print (
    std::ostream & os) const
```

12.245.4.11 Read()

```
std::istream & gdcmm::network::PresentationContextRQ::Read (
    std::istream & is)
```

12.245.4.12 SetAbstractSyntax()

```
void gdcmm::network::PresentationContextRQ::SetAbstractSyntax (
    AbstractSyntax const & absyn)
```

12.245.4.13 SetPresentationContextID()

```
void gdcmm::network::PresentationContextRQ::SetPresentationContextID (
    uint8_t id)
```

12.245.4.14 Size()

```
size_t gdcmm::network::PresentationContextRQ::Size () const
```


12.245.4.15 Write()

```
const std::ostream & gdcm::network::PresentationContextRQ::Write (
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmPresentationContextRQ.h](#)

12.246 gdcm::network::PresentationDataValue Class Reference

[PresentationDataValue](#).

```
#include <gdcmPresentationDataValue.h>
```

Public Member Functions

- [PresentationDataValue](#) ()
- const std::string & [GetBlob](#) () const
- bool [GetIsCommand](#) () const
- bool [GetIsLastFragment](#) () const
- uint8_t [GetMessageHeader](#) () const
- uint8_t [GetPresentationContextID](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- std::istream & [ReadInto](#) (std::istream &is, std::ostream &os)
- void [SetBlob](#) (const std::string &partialblob)
- void [SetCommand](#) (bool inCommand)
- void [SetDataSet](#) (const [DataSet](#) &ds)
- void [SetLastFragment](#) (bool inLast)
- void [SetMessageHeader](#) (uint8_t messageheader)
- void [SetPresentationContextID](#) (uint8_t id)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

Static Public Member Functions

- static [DataSet](#) [ConcatenatePDVBlobs](#) (const std::vector< [PresentationDataValue](#) > &inPDVs)
- static [DataSet](#) [ConcatenatePDVBlobsAsExplicit](#) (const std::vector< [PresentationDataValue](#) > &inPDVs)

12.246.1 Detailed Description

[PresentationDataValue](#).

[Table 9-23](#) PRESENTATION-DATA-VALUE ITEM FIELDS

12.246.2 Constructor & Destructor Documentation

12.246.2.1 PresentationDataValue()

gdcm::network::PresentationDataValue::PresentationDataValue ()

12.246.3 Member Function Documentation

12.246.3.1 ConcatenatePDVBlobs()

[DataSet](#) gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (
const std::vector< [PresentationDataValue](#) > & inPDVs) [static]

Warning

[DataSet](#) will be read as Implicit Little Endian TS

12.246.3.2 ConcatenatePDVBlobsAsExplicit()

[DataSet](#) gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
const std::vector< [PresentationDataValue](#) > & inPDVs) [static]

12.246.3.3 GetBlob()

const std::string & gdcm::network::PresentationDataValue::GetBlob () const

12.246.3.4 GetIsCommand()

bool gdcm::network::PresentationDataValue::GetIsCommand () const

12.246.3.5 GetIsLastFragment()

bool gdcm::network::PresentationDataValue::GetIsLastFragment () const

12.246.3.6 GetMessageHeader()

uint8_t gdcm::network::PresentationDataValue::GetMessageHeader () const [inline]

References [gdcm_assert](#).

12.246.3.7 GetPresentationContextID()

```
uint8_t gdcmm::network::PresentationDataValue::GetPresentationContextID () const [inline]
```

12.246.3.8 Print()

```
void gdcmm::network::PresentationDataValue::Print (  
    std::ostream & os) const
```

12.246.3.9 Read()

```
std::istream & gdcmm::network::PresentationDataValue::Read (  
    std::istream & is)
```

12.246.3.10 ReadInto()

```
std::istream & gdcmm::network::PresentationDataValue::ReadInto (  
    std::istream & is,  
    std::ostream & os)
```

12.246.3.11 SetBlob()

```
void gdcmm::network::PresentationDataValue::SetBlob (  
    const std::string & partialblob)
```

12.246.3.12 SetCommand()

```
void gdcmm::network::PresentationDataValue::SetCommand (  
    bool inCommand)
```

12.246.3.13 SetDataSet()

```
void gdcmm::network::PresentationDataValue::SetDataSet (  
    const DataSet & ds)
```

Set [DataSet](#). Write [DataSet](#) in implicit.

Warning

size of dataset should be below maxpdusize

12.246.3.14 SetLastFragment()

```
void gdcmm::network::PresentationDataValue::SetLastFragment (  
    bool inLast)
```

12.246.3.15 SetMessageHeader()

```
void gdcmm::network::PresentationDataValue::SetMessageHeader (  
    uint8_t messageheader) [inline]
```

References [gdcmm_assert](#).

12.246.3.16 SetPresentationContextID()

```
void gdcmm::network::PresentationDataValue::SetPresentationContextID (  
    uint8_t id) [inline]
```

References [gdcmm_assert](#).

12.246.3.17 Size()

```
size_t gdcmm::network::PresentationDataValue::Size () const
```

12.246.3.18 Write()

```
const std::ostream & gdcmm::network::PresentationDataValue::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

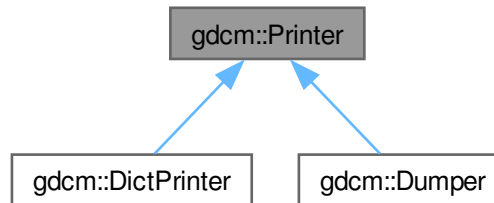
- [gdcmmPresentationDataValue.h](#)

12.247 gdcmm::Printer Class Reference

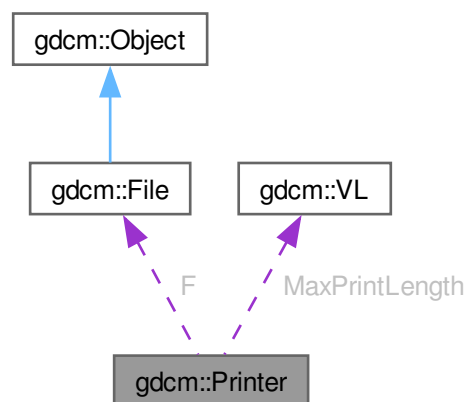
[Printer](#) class.

```
#include <gdcmmPrinter.h>
```

Inheritance diagram for gdcmm::Printer:



Collaboration diagram for gdcmm::Printer:



Public Types

- enum [PrintStyles](#) {
 [VERBOSE_STYLE](#) = 0 ,
 [CONDENSED_STYLE](#) ,
 [XML](#) ,
 [CXX](#) }

Public Member Functions

- [Printer](#) ()
- [~Printer](#) ()=default
- [PrintStyles](#) [GetPrintStyle](#) () const
Get PrintStyle value.
- void [Print](#) (std::ostream &os)
Print.
- void [PrintDataSet](#) (const [DataSet](#) &ds, std::ostream &os, const std::string &rs="")
Print an individual dataset.
- void [SetColor](#) (bool c)
Set color mode or not.
- void [SetFile](#) ([File](#) const &f)
Set file.
- void [SetStyle](#) ([PrintStyles](#) ps)
Set PrintStyle value.

Protected Member Functions

- [VR](#) [PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, std::ostream &out, std::string const &indent)
- void [PrintSQ](#) (const [SequenceOfItems](#) *sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const [File](#) * F
- [VL](#) [MaxPrintLength](#)
- [PrintStyles](#) [PrintStyle](#)

12.247.1 Detailed Description

[Printer](#) class.

Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

12.247.2 Member Enumeration Documentation

12.247.2.1 PrintStyles

enum [gdcm::Printer::PrintStyles](#)

Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

12.247.3 Constructor & Destructor Documentation

12.247.3.1 Printer()

`gdcm::Printer::Printer ()`

12.247.3.2 ~Printer()

`gdcm::Printer::~~Printer ()` [default]

12.247.4 Member Function Documentation

12.247.4.1 GetPrintStyle()

[PrintStyles](#) `gdcm::Printer::GetPrintStyle () const` [inline]

Get PrintStyle value.

References [PrintStyle](#).

12.247.4.2 Print()

`void gdcm::Printer::Print (
 std::ostream & os)`

Print.

Examples

[DumpSiemensBase64.cxx](#).

12.247.4.3 PrintDataElement()

[VR](#) `gdcm::Printer::PrintDataElement (`
 `std::ostream & os,`
 `const Dicts & dicts,`
 `const DataSet & ds,`
 `const DataElement & de,`
 `std::ostream & out,`
 `std::string const & indent) [protected]`

12.247.4.4 PrintDataSet()

`void gdcm::Printer::PrintDataSet (`
 `const DataSet & ds,`
 `std::ostream & os,`
 `const std::string & s = "")`

Print an individual dataset.

12.247.4.5 PrintSQ()

`void gdcm::Printer::PrintSQ (`
 `const SequenceOfItems * sqi,`
 `std::ostream & os,`
 `std::string const & indent) [protected]`

12.247.4.6 SetColor()

`void gdcm::Printer::SetColor (`
 `bool c)`

Set color mode or not.

12.247.4.7 SetFile()

`void gdcm::Printer::SetFile (`
 `File const & f) [inline]`

Set file.

Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), and [DumpToshibaDTI2.cxx](#).

References [F](#).

12.247.4.8 SetStyle()

```
void gdcm::Printer::SetStyle (
    PrintStyles ps) [inline]
```

Set [PrintStyle](#) value.

References [PrintStyle](#).

12.247.5 Member Data Documentation

12.247.5.1 F

```
const File* gdcm::Printer::F [protected]
```

Referenced by [SetFile\(\)](#).

12.247.5.2 MaxPrintLength

```
VL gdcm::Printer::MaxPrintLength [protected]
```

12.247.5.3 PrintStyle

```
PrintStyles gdcm::Printer::PrintStyle [protected]
```

Referenced by [gdcm::Dumper::Dumper\(\)](#), [GetPrintStyle\(\)](#), and [SetStyle\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmPrinter.h](#)

12.248 gdcm::PrivateDict Class Reference

Private [Dict](#).

```
#include <gdcmDict.h>
```

Public Member Functions

- [PrivateDict](#) ()=default
- [~PrivateDict](#) ()=default
- void [AddDictEntry](#) (const [PrivateTag](#) &tag, const [DictEntry](#) &de)
- bool [FindDictEntry](#) (const [PrivateTag](#) &tag) const
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- bool [IsEmpty](#) () const
- void [PrintXML](#) () const
- bool [RemoveDictEntry](#) (const [PrivateTag](#) &tag)

Protected Member Functions

- void [LoadDefault](#) ()

Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &os, const [PrivateDict](#) &val)

12.248.1 Detailed Description

Private [Dict](#).

12.248.2 Constructor & Destructor Documentation

12.248.2.1 PrivateDict()

gdcmm::PrivateDict::PrivateDict () [default]

Referenced by [LoadDefault\(\)](#), and [operator<<](#).

12.248.2.2 ~PrivateDict()

gdcmm::PrivateDict::~~PrivateDict () [default]

12.248.3 Member Function Documentation

12.248.3.1 AddDictEntry()

```
void gdcmm::PrivateDict::AddDictEntry (
    const PrivateTag & tag,
    const DictEntry & de) [inline]
```

References [gdcmm_assert](#), [GetDictEntry\(\)](#), [gdcmm::DictEntry::GetVM\(\)](#), [gdcmm::DictEntry::GetVR\(\)](#), [gdcmm::DictEntry::SetVM\(\)](#), [gdcmm::DictEntry::SetVR\(\)](#), and [gdcmm::VR::UN](#).

12.248.3.2 FindDictEntry()

```
bool gdcmm::PrivateDict::FindDictEntry (
    const PrivateTag & tag) const [inline]
```

12.248.3.3 GetDictEntry()

```
const DictEntry & gdcm::PrivateDict::GetDictEntry (  
    const PrivateTag & tag) const    [inline]
```

References [gdcm_assert](#).

Referenced by [AddDictEntry\(\)](#).

12.248.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty () const    [inline]
```

12.248.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ()    [protected]
```

References [PrivateDict\(\)](#).

12.248.3.6 PrintXML()

```
void gdcm::PrivateDict::PrintXML () const    [inline]
```

References [gdcm::Tag::GetElement\(\)](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DictEntry::GetName\(\)](#), [gdcm::PrivateTag::GetOwner\(\)](#), [gdcm::DictEntry::GetVM\(\)](#), and [gdcm::DictEntry::GetVR\(\)](#).

12.248.3.7 RemoveDictEntry()

```
bool gdcm::PrivateDict::RemoveDictEntry (  
    const PrivateTag & tag)    [inline]
```

Remove entry 'tag'. Return true on success (element was found and remove). return false if element was not found.

References [gdcm_assert](#).

12.248.4 Friends And Related Symbol Documentation

12.248.4.1 Dicts

```
friend class Dicts    [friend]
```

References [Dicts](#).

Referenced by [Dicts](#).

12.248.4.2 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const PrivateDict & val) [friend]
```

References [PrivateDict\(\)](#).

The documentation for this class was generated from the following file:

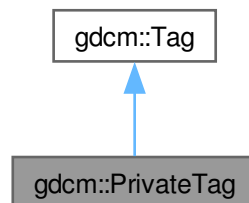
- [gdcmDict.h](#)

12.249 gdcm::PrivateTag Class Reference

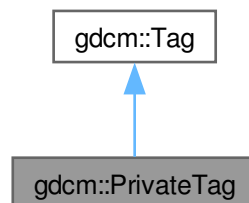
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner).

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- [PrivateTag](#) ([Tag](#) const &t, const char *owner="")
- [PrivateTag](#) (uint16_t group=0, uint16_t element=0, const char *owner="")
- [DataElement](#) [GetAsDataElement](#) () const
- const char * [GetOwner](#) () const
- bool [operator!=](#) (const [PrivateTag](#) &_val) const
- bool [operator!=](#) (const [Tag](#) &_val) const
- bool [operator<](#) (const [PrivateTag](#) &_val) const
- [PrivateTag](#) & [operator=](#) (const [PrivateTag](#) &_val)
- bool [operator==](#) (const [PrivateTag](#) &_val) const
- bool [operator==](#) (const [Tag](#) &_val) const
- bool [ReadFromCommaSeparatedString](#) (const char *str)
- void [SetOwner](#) (const char *owner)

Public Member Functions inherited from [gdcm::Tag](#)

- [Tag](#) (const [Tag](#) &_val)
- [Tag](#) (uint16_t group, uint16_t element)
Constructor with 2*uint16_t.
- [Tag](#) (uint32_t tag=0)
Constructor with 1*uint32_t Prefer the ctor that takes two uint16_t.
- uint16_t [GetElement](#) () const
Returns the 'Element number' of the given [Tag](#).
- uint32_t [GetElementTag](#) () const
Returns the full tag value of the given [Tag](#).
- uint16_t [GetGroup](#) () const
Returns the 'Group number' of the given [Tag](#).
- uint32_t [GetLength](#) () const
return the length of tag (read: size on disk)
- [Tag](#) [GetPrivateCreator](#) () const
Return the Private Creator Data [Element](#) tag of a private data element.
- bool [IsGroupLength](#) () const
return whether the tag correspond to a group length tag:
- bool [IsGroupXX](#) (const [Tag](#) &t) const
e.g 6002,3000 belong to groupXX: 6000,3000
- bool [IsIllegal](#) () const
return if the tag is considered to be an illegal tag
- bool [IsPrivate](#) () const
- bool [IsPrivateCreator](#) () const
- bool [IsPublic](#) () const
- bool [operator!=](#) (const [Tag](#) &_val) const
- bool [operator<](#) (const [Tag](#) &_val) const
- bool [operator<=](#) (const [Tag](#) &t2) const
- [Tag](#) & [operator=](#) (const [Tag](#) &_val)
- bool [operator==](#) (const [Tag](#) &_val) const
- uint16_t & [operator\[\]](#) (const unsigned int &_id)
Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).

- `const uint16_t & operator[] (const unsigned int &_id) const`
Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).
- `std::string PrintAsContinuousString () const`
- `std::string PrintAsContinuousUpperCaseString () const`
Same as `PrintAsContinuousString`, but hexadecimal [a-f] are printed using upper case.
- `std::string PrintAsPipeSeparatedString () const`
- `template<typename TSwap>`
`std::istream & Read (std::istream &is)`
Read a tag from binary representation.
- `bool ReadFromCommaSeparatedString (const char *str)`
- `bool ReadFromContinuousString (const char *str)`
- `bool ReadFromPipeSeparatedString (const char *str)`
- `void SetElement (uint16_t element)`
Sets the '[Element](#) number' of the given [Tag](#).
- `void SetElementTag (uint16_t group, uint16_t element)`
Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).
- `void SetElementTag (uint32_t tag)`
Sets the full tag value of the given [Tag](#).
- `void SetGroup (uint16_t group)`
Sets the 'Group number' of the given [Tag](#).
- `void SetPrivateCreator (Tag const &t)`
Set private creator:
- `template<typename TSwap>`
`const std::ostream & Write (std::ostream &os) const`
Write a tag in binary rep.

Friends

- `std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)`

12.249.1 Detailed Description

Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner).

Note

private tag have element value in: [0x10,0xff], for instance 0x0009,0x0000 is NOT a private tag

Examples

[ChangePrivateTags.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [ELSCINT1WaveToText.cxx](#), [FileStreaming.cs](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadGEMSSDO.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.249.2 Constructor & Destructor Documentation

12.249.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (  
    uint16_t group = 0,  
    uint16_t element = 0,  
    const char * owner = "") [inline]
```

References [gdcm::Tag::Tag\(\)](#), and [gdcm::Tag::SetElement\(\)](#).

Referenced by [operator!=\(\)](#), [operator<\(\)](#), [operator<<\(\)](#), [operator=\(\)](#), and [operator==\(\)](#).

12.249.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (  
    Tag const & t,  
    const char * owner = "") [inline]
```

References [gdcm::Tag::Tag\(\)](#), [gdcm::Tag::GetElement\(\)](#), and [gdcm::Tag::SetElement\(\)](#).

12.249.3 Member Function Documentation

12.249.3.1 GetAsDataElement()

[DataElement](#) gdcm::PrivateTag::GetAsDataElement () const

12.249.3.2 GetOwner()

const char * gdcm::PrivateTag::GetOwner () const [inline]

Examples

[PublicDict.cxx](#).

Referenced by [gdcm::PrivateDict::PrintXML\(\)](#).

12.249.3.3 operator"!=()" [1/2]

```
bool gdcm::PrivateTag::operator!= (  
    const PrivateTag & __val) const [inline]
```

References [PrivateTag\(\)](#), and [gdcm::Tag::GetElementTag\(\)](#).

12.249.3.4 `operator"!==()` [2/2]

```
bool gdcm::PrivateTag::operator!=(
    const Tag & _val) const    [inline]
```

References [gdcm::Tag::Tag\(\)](#), and [gdcm::Tag::GetElementTag\(\)](#).

12.249.3.5 `operator<()`

```
bool gdcm::PrivateTag::operator< (
    const PrivateTag & _val) const
```

References [PrivateTag\(\)](#).

12.249.3.6 `operator==()`

```
PrivateTag & gdcm::PrivateTag::operator= (
    const PrivateTag & _val)    [inline]
```

References [PrivateTag\(\)](#), [gdcm::Tag::GetElementTag\(\)](#), and [gdcm::Tag::SetElementTag\(\)](#).

12.249.3.7 `operator==()` [1/2]

```
bool gdcm::PrivateTag::operator==(
    const PrivateTag & _val) const    [inline]
```

References [PrivateTag\(\)](#), and [gdcm::Tag::GetElementTag\(\)](#).

12.249.3.8 `operator==()` [2/2]

```
bool gdcm::PrivateTag::operator==(
    const Tag & _val) const    [inline]
```

References [gdcm::Tag::Tag\(\)](#), and [gdcm::Tag::GetElementTag\(\)](#).

12.249.3.9 `ReadFromCommaSeparatedString()`

```
bool gdcm::PrivateTag::ReadFromCommaSeparatedString (
    const char * str)
```

Read [PrivateTag](#) from a string. [Element](#) number will be truncated to 8bits. Eg: "1234,5678,GDCM" is private tag: (1234,78,"GDCM")

12.249.3.10 SetOwner()

```
void gdcm::PrivateTag::SetOwner (  
    const char * owner) [inline]
```

References [gdcm::String<'\\', 64 >::Trim\(\)](#).

12.249.4 Friends And Related Symbol Documentation

12.249.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const PrivateTag & __val) [friend]
```

References [PrivateTag\(\)](#), and [operator<<](#).

Referenced by [operator<<](#).

The documentation for this class was generated from the following file:

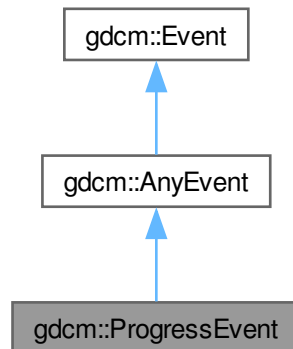
- [gdcmPrivateTag.h](#)

12.250 gdcm::ProgressEvent Class Reference

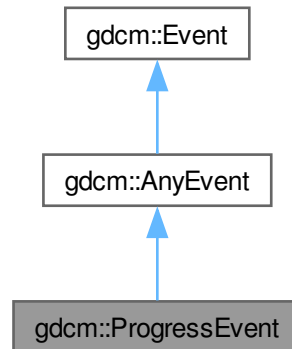
[ProgressEvent](#).

```
#include <gdcmProgressEvent.h>
```

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for `gdcm::ProgressEvent`:



Public Types

- typedef [ProgressEvent](#) `Self`
- typedef [AnyEvent](#) `Superclass`

Public Member Functions

- [ProgressEvent](#) (`const Self &s`)
- [ProgressEvent](#) (`double p=0`)
- [~ProgressEvent](#) () `override=default`
- `bool` [CheckEvent](#) (`const ::gdcm::Event *e`) `const override`
- `const char *` [GetEventName](#) () `const override`
- `double` [GetProgress](#) () `const`
- `::gdcm::Event *` [MakeObject](#) () `const override`
- `void` [operator=](#) (`const Self &`)`=delete`
- `void` [SetProgress](#) (`double p`)

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (`const Event &`)
- `virtual` [~Event](#) ()
- `virtual bool` [CheckEvent](#) (`const Event *`) `const =0`
- `void` [operator=](#) (`const Event &`)`=delete`
- `virtual void` [Print](#) (`std::ostream &os`) `const`

12.250.1 Detailed Description

[ProgressEvent](#).

Special type of event triggered during

See also

[AnyEvent](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.250.2 Member Typedef Documentation

12.250.2.1 Self

```
typedef ProgressEvent gdcmm::ProgressEvent::Self
```

12.250.2.2 Superclass

```
typedef AnyEvent gdcmm::ProgressEvent::Superclass
```

12.250.3 Constructor & Destructor Documentation

12.250.3.1 ProgressEvent() [1/2]

```
gdcmm::ProgressEvent::ProgressEvent (  
    double p = 0)    [inline]
```

12.250.3.2 ~ProgressEvent()

```
gdcmm::ProgressEvent::~~ProgressEvent ()    [override], [default]
```

12.250.3.3 ProgressEvent() [2/2]

```
gdcmm::ProgressEvent::ProgressEvent (  
    const Self & s)    [inline]
```

12.250.4 Member Function Documentation

12.250.4.1 CheckEvent()

```
bool gdcmm::ProgressEvent::CheckEvent (  
    const gdcmm::Event * e) const    [inline], [override]
```

12.250.4.2 GetEventName()

```
const char * gdcmm::ProgressEvent::GetEventName () const    [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

12.250.4.3 GetProgress()

```
double gdcmm::ProgressEvent::GetProgress () const    [inline]
```

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

12.250.4.4 MakeObject()

```
gdcmm::Event * gdcmm::ProgressEvent::MakeObject () const    [inline], [override], [virtual]
```

Create an [Event](#) of this type This method work as a Factory for creating events of each particular type.

Implements [gdcmm::Event](#).

12.250.4.5 operator=()

```
void gdcmm::ProgressEvent::operator= (  
    const Self & )    [delete]
```

12.250.4.6 SetProgress()

```
void gdcmm::ProgressEvent::SetProgress (  
    double p)    [inline]
```

The documentation for this class was generated from the following file:

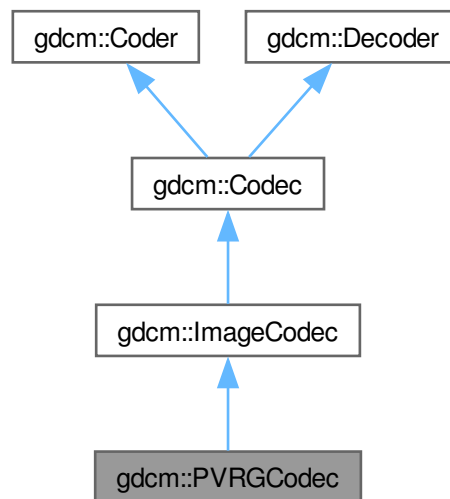
- [gdcmmProgressEvent.h](#)

12.251 gdcm::PVRGCodec Class Reference

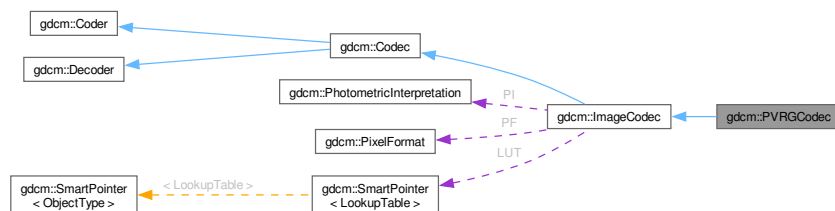
[PVRGCodec](#).

```
#include <gdcmPVRGCodec.h>
```

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- [PVRGCodec](#) ()
- [~PVRGCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override

- Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
- Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
- Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
- Decode.
- void [SetLossyFlag](#) (bool l)

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- virtual bool [GetHeaderInfo](#) (std::istream &is_, [TransferSyntax](#) &ts)
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DecodeByStreams](#) (std::istream &is_, std::ostream &os) override
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.251.1 Detailed Description

[PVRGCodec](#).

Note

pvrp is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS_Gyrosan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

12.251.2 Constructor & Destructor Documentation

12.251.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ()
```

12.251.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~~PVRGCodec () [override]
```

12.251.3 Member Function Documentation

12.251.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.251.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.251.3.3 Clone()

[ImageCodec](#) * gdcm::PVRGCodec::Clone () const [override], [virtual]

Implements [gdcm::ImageCodec](#).

References [gdcm::ImageCodec::ImageCodec\(\)](#).

12.251.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
    DataElement const & in_,
    DataElement & out_) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

12.251.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

12.251.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
    bool l)
```

The documentation for this class was generated from the following file:

- [gdcmPVRGCodec.h](#)

12.252 gdcm::PythonFilter Class Reference

[PythonFilter](#) [PythonFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmPythonFilter.h>
```

Public Member Functions

- [PythonFilter](#) ()
- [~PythonFilter](#) ()
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetDicts](#) (const [Dicts](#) &dicts)
- void [SetFile](#) (const [File](#) &f)
- PyObject * [ToPyObject](#) (const [Tag](#) &t) const
- void [UseDictAlways](#) (bool)

12.252.1 Detailed Description

[PythonFilter](#) [PythonFilter](#) is the class that make gdc2.x looks more like gdc1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

12.252.2 Constructor & Destructor Documentation

12.252.2.1 PythonFilter()

gdc2::PythonFilter::PythonFilter ()

12.252.2.2 ~PythonFilter()

gdc2::PythonFilter::~~PythonFilter ()

12.252.3 Member Function Documentation

12.252.3.1 GetFile() [1/2]

[File](#) & gdc2::PythonFilter::GetFile ()

12.252.3.2 GetFile() [2/2]

const [File](#) & gdc2::PythonFilter::GetFile () const

12.252.3.3 SetDicts()

void gdc2::PythonFilter::SetDicts (
const [Dicts](#) & dicts)

12.252.3.4 SetFile()

```
void gdcm::PythonFilter::SetFile (
    const File & f)
```

12.252.3.5 ToPyObject()

```
PyObject * gdcm::PythonFilter::ToPyObject (
    const Tag & t) const
```

12.252.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
    bool ) [inline]
```

The documentation for this class was generated from the following file:

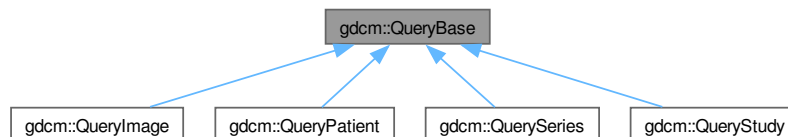
- [gdcmPythonFilter.h](#)

12.253 gdcm::QueryBase Class Reference

[QueryBase](#).

```
#include <gdcmQueryBase.h>
```

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const
- virtual std::vector< [Tag](#) > [GetHierachicalSearchTags](#) (const [ERootType](#) &inRootType) const =0
Return all Unique Key for a particular Query Root type (from the same level and above).
- virtual const char * [GetName](#) () const =0
- virtual std::vector< [Tag](#) > [GetOptionalTags](#) (const [ERootType](#) &inRootType) const =0
- virtual [DataElement](#) [GetQueryLevel](#) () const =0
- virtual std::vector< [Tag](#) > [GetRequiredTags](#) (const [ERootType](#) &inRootType) const =0
- virtual std::vector< [Tag](#) > [GetUniqueTags](#) (const [ERootType](#) &inRootType) const =0

12.253.1 Detailed Description

[QueryBase](#).

contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE

There are four levels of C-FIND and C-MOVE query:

- [Patient](#)
- [Study](#)
- [Series](#)
- [Image](#)

Each one has its own required and optional tags. This class provides an interface for getting those tags. This is an interface class.

See 3.4 C 6.1 and 3.4 C 6.2 for the patient and study root query types. These sections define the tags allowed by a particular query. The caller must pass in which root type they want, patient or study. A third root type, Modality Worklist Query, isn't yet supported.

This class (or rather it's derived classes) will be held in the RootQuery types. These query types actually make the dataset, and will use this dataset to list the required, unique, and optional tags for each type of query. This design is somewhat overly complicated, but is kept so that if we ever wanted to try to guess the query type from the given tags, we could do so.

12.253.2 Constructor & Destructor Documentation

12.253.2.1 [~QueryBase\(\)](#)

```
virtual gdcmm::QueryBase::~QueryBase () [virtual], [default]
```

12.253.3 Member Function Documentation

12.253.3.1 [GetAllRequiredTags\(\)](#)

```
std::vector< Tag > gdcmm::QueryBase::GetAllRequiredTags (
    const ERootType & inRootType) const
```

In order to validate a query dataset we need to check that there exists at least one required (or unique) key

12.253.3.2 [GetAllTags\(\)](#)

```
std::vector< Tag > gdcmm::QueryBase::GetAllTags (
    const ERootType & inRootType) const
```

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

12.253.3.3 GetHierarchicalSearchTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetHierarchicalSearchTags (  
    const ERootType & inRootType) const    [pure virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

12.253.3.4 GetName()

```
virtual const char * gdcM::QueryBase::GetName () const    [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

12.253.3.5 GetOptionalTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetOptionalTags (  
    const ERootType & inRootType) const    [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

12.253.3.6 GetQueryLevel()

```
virtual DataElement gdcM::QueryBase::GetQueryLevel () const    [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

12.253.3.7 GetRequiredTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetRequiredTags (  
    const ERootType & inRootType) const    [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

12.253.3.8 GetUniqueTags()

```
virtual std::vector< Tag > gdcM::QueryBase::GetUniqueTags (  
    const ERootType & inRootType) const    [pure virtual]
```

Implemented in [gdcM::QueryImage](#), [gdcM::QueryPatient](#), [gdcM::QuerySeries](#), and [gdcM::QueryStudy](#).

The documentation for this class was generated from the following file:

- [gdcMQueryBase.h](#)

12.254 gdcm::QueryFactory Class Reference

QueryFactory.h.

```
#include <gdcmQueryFactory.h>
```

Static Public Member Functions

- static [ECharSet](#) [GetCharacterFromCurrentLocale](#) ()
- static void [ListCharSets](#) (std::ostream &os)
List all possible CharSet.
- static [DataElement](#) [ProduceCharacterSetDataElement](#) (const std::vector< [ECharSet](#) > &inCharSet, [Type](#))
- static [BaseQuery](#) * [ProduceQuery](#) (const std::string &sopInstanceUID, [ENQueryType](#) inQueryType)
- static [BaseRootQuery](#) * [ProduceQuery](#) ([ERootType](#) inRootType, [EQueryType](#) inQueryType, [EQueryLevel](#) inQueryLevel)

12.254.1 Detailed Description

QueryFactory.h.

contains: a class to produce a query based off of user-entered information

Essentially, this class is used to construct a query based off of user input (typically from the command line; if in code directly, the query itself could just be instantiated)

In theory, could also be used as the interface to validate incoming datasets as belonging to a particular query style

12.254.2 Member Function Documentation

12.254.2.1 GetCharacterFromCurrentLocale()

```
ECharSet gdcm::QueryFactory::GetCharacterFromCurrentLocale () [static]
```

This function will return the corresponding [ECharSet](#) associated with the current locale of the running system (based on the value of locale()).

12.254.2.2 ListCharSets()

```
void gdcm::QueryFactory::ListCharSets (  
    std::ostream & os) [static]
```

List all possible CharSet.

12.254.2.3 ProduceCharacterSetDataElement()

```
DataElement gdcm::QueryFactory::ProduceCharacterSetDataElement (
    const std::vector< ECharSet > & inCharSetType) [static]
```

This function will produce the appropriate dataelement given a list of charsets. The first charset will be used directly, while the second and subsequent will be prepended with "ISO2022 ". Redundant character sets are not permitted, so if they are encountered, they will just be skipped. if UTF8 or GB18030 is used, no subsequent character sets will be used if the vector passed in is empty, then the dataelement that's passed out will be empty and Latin1 is the presumed encoding

12.254.2.4 ProduceQuery() [1/2]

```
BaseQuery * gdcm::QueryFactory::ProduceQuery (
    const std::string & sopInstanceUID,
    ENQueryType inQueryType) [static]
```

12.254.2.5 ProduceQuery() [2/2]

```
BaseRootQuery * gdcm::QueryFactory::ProduceQuery (
    ERootType inRootType,
    EQueryType inQueryType,
    EQueryLevel inQueryLevel) [static]
```

this function will produce a query (basically, a wrapper to a dataset that can validate whether or not the query is a valid cfind/cmove query) and the level of the query (patient, study, series, image). If the user provides an invalid instantiation (ie, study root type, query level of patient), then the result is NULL.

The documentation for this class was generated from the following file:

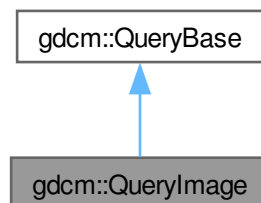
- [gdcmQueryFactory.h](#)

12.255 gdcm::QueryImage Class Reference

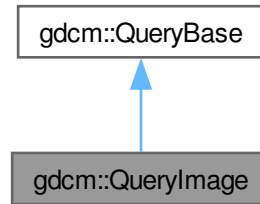
[QueryImage](#).

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for `gdcm::QueryImage`:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &`inRootType`) const override
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const override

Public Member Functions inherited from `gdcm::QueryBase`

- `virtual ~QueryBase` ()=default
- `std::vector< Tag > GetAllRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetAllTags` (const `ERootType` &`inRootType`) const

12.255.1 Detailed Description

`QueryImage`.

contains: class to construct an image-based query for C-FIND and C-MOVE

12.255.2 Member Function Documentation

12.255.2.1 `GetHierachicalSearchTags()`

```
std::vector< Tag > gdcm::QueryImage::GetHierachicalSearchTags (
    const ERootType & inRootType) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements `gdcm::QueryBase`.

12.255.2.2 GetName()

```
const char * gdcm::QueryImage::GetName () const  [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.255.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryImage::GetOptionalTags (  
    const ERootType & inRootType) const  [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.255.2.4 GetQueryLevel()

```
DataElement gdcm::QueryImage::GetQueryLevel () const  [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.255.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryImage::GetRequiredTags (  
    const ERootType & inRootType) const  [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.255.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryImage::GetUniqueTags (  
    const ERootType & inRootType) const  [override], [virtual]
```

Implements [gdcm::QueryBase](#).

The documentation for this class was generated from the following file:

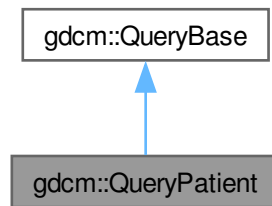
- [gdcmQueryImage.h](#)

12.256 gdcM::QueryPatient Class Reference

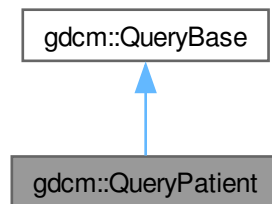
[QueryPatient](#).

```
#include <gdcMQueryPatient.h>
```

Inheritance diagram for gdcM::QueryPatient:



Collaboration diagram for gdcM::QueryPatient:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const [ERootType](#) &inRootType) const override
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const [ERootType](#) &inRootType) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const [ERootType](#) &inRootType) const override
- `std::vector< Tag > GetUniqueTags` (const [ERootType](#) &inRootType) const override

Public Member Functions inherited from [gdcm::QueryBase](#)

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const

12.256.1 Detailed Description

[QueryPatient](#).

contains: class to construct a patient-based query for c-find and c-move

12.256.2 Member Function Documentation

12.256.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryPatient::GetHierachicalSearchTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

12.256.2.2 GetName()

```
const char * gdcm::QueryPatient::GetName () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.256.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryPatient::GetOptionalTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.256.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.256.2.5 GetRequiredTags()

```
std::vector< Tag > gdcM::QueryPatient::GetRequiredTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcM::QueryBase](#).

12.256.2.6 GetUniqueTags()

```
std::vector< Tag > gdcM::QueryPatient::GetUniqueTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcM::QueryBase](#).

The documentation for this class was generated from the following file:

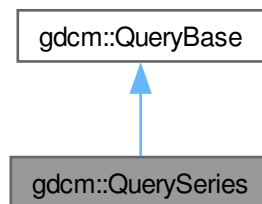
- [gdcMQueryPatient.h](#)

12.257 gdcM::QuerySeries Class Reference

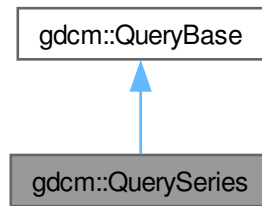
[QuerySeries](#).

```
#include <gdcMQuerySeries.h>
```

Inheritance diagram for gdcM::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &`inRootType`) const override
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const override

Public Member Functions inherited from `gdcm::QueryBase`

- `virtual ~QueryBase` ()=default
- `std::vector< Tag > GetAllRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetAllTags` (const `ERootType` &`inRootType`) const

12.257.1 Detailed Description

`QuerySeries`.

contains: class to construct a series-based query for c-find and c-move

12.257.2 Member Function Documentation

12.257.2.1 `GetHierachicalSearchTags()`

```
std::vector< Tag > gdcm::QuerySeries::GetHierachicalSearchTags (
    const ERootType & inRootType) const [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements `gdcm::QueryBase`.

12.257.2.2 GetName()

```
const char * gdcm::QuerySeries::GetName () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.257.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QuerySeries::GetOptionalTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.257.2.4 GetQueryLevel()

```
DataElement gdcm::QuerySeries::GetQueryLevel () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.257.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QuerySeries::GetRequiredTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.257.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QuerySeries::GetUniqueTags (  
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

The documentation for this class was generated from the following file:

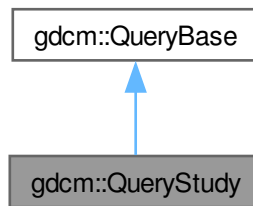
- [gdcmQuerySeries.h](#)

12.258 gdcm::QueryStudy Class Reference

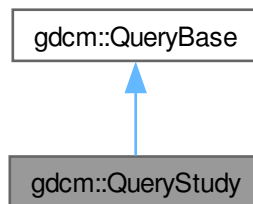
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags` (const `ERootType` &inRootType) const override
Return all Unique Key for a particular Query Root type (from the same level and above).
- `const char * GetName` () const override
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &inRootType) const override
- `DataElement GetQueryLevel` () const override
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &inRootType) const override
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &inRootType) const override

Public Member Functions inherited from [gdcm::QueryBase](#)

- virtual [~QueryBase](#) ()=default
- std::vector< [Tag](#) > [GetAllRequiredTags](#) (const [ERootType](#) &inRootType) const
- std::vector< [Tag](#) > [GetAllTags](#) (const [ERootType](#) &inRootType) const

12.258.1 Detailed Description

QueryStudy.h.

contains: class to construct a study-based query for C-FIND and C-MOVE

12.258.2 Member Function Documentation

12.258.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryStudy::GetHierachicalSearchTags (
    const ERootType & inRootType) const    [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

12.258.2.2 GetName()

```
const char * gdcm::QueryStudy::GetName () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.258.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryStudy::GetOptionalTags (
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.258.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel () const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.258.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryStudy::GetRequiredTags (
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

12.258.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryStudy::GetUniqueTags (
    const ERootType & inRootType) const    [override], [virtual]
```

Implements [gdcm::QueryBase](#).

The documentation for this class was generated from the following file:

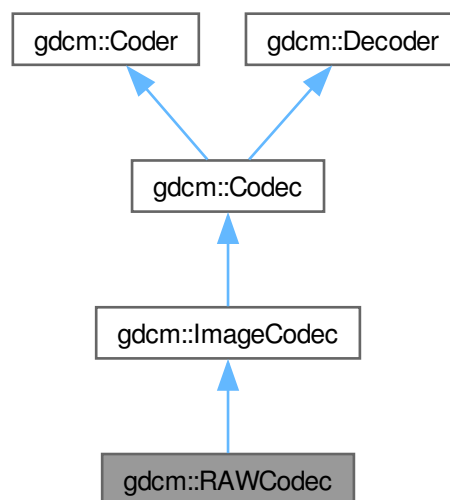
- [gdcmQueryStudy.h](#)

12.259 gdcm::RAWCodec Class Reference

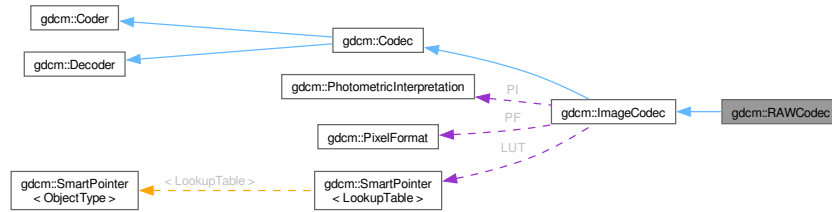
[RAWCodec](#) class.

```
#include <gdcmRAWCodec.h>
```

Inheritance diagram for `gdcm::RAWCodec`:



Collaboration diagram for `gdcm::RAWCodec`:



Public Member Functions

- [RAWCodec](#) ()
- [~RAWCodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- bool [DecodeBytes](#) (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)↵
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override

Public Member Functions inherited from [gdcm::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])
- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)

- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- virtual bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen)
- virtual bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen)
- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsFrameEncoder](#) ()
- virtual bool [IsRowEncoder](#) ()
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- virtual bool [StartEncode](#) (std::ostream &os)
- virtual bool [StopEncode](#) (std::ostream &os)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) LUT
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) PF
- [PhotometricInterpretation](#) PI
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.259.1 Detailed Description

[RAWCodec](#) class.

12.259.2 Constructor & Destructor Documentation

12.259.2.1 RAWCodec()

`gdcm::RAWCodec::RAWCodec ()`

12.259.2.2 ~RAWCodec()

`gdcm::RAWCodec::~~RAWCodec ()` [override]

12.259.3 Member Function Documentation

12.259.3.1 CanCode()

`bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const` [override], [virtual]

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcm::ImageCodec](#).

12.259.3.2 CanDecode()

`bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const` [override], [virtual]

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcm::ImageCodec](#).

12.259.3.3 Clone()

[ImageCodec](#) * gdcmm::RAWCodec::Clone () const [override], [virtual]

Implements [gdcmm::ImageCodec](#).

References [gdcmm::ImageCodec::ImageCodec\(\)](#).

12.259.3.4 Code()

```
bool gdcmm::RAWCodec::Code (  
    DataElement const & in_,  
    DataElement & out_) [override], [virtual]
```

Code.

Reimplemented from [gdcmm::Coder](#).

12.259.3.5 Decode()

```
bool gdcmm::RAWCodec::Decode (  
    DataElement const & ,  
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcmm::ImageCodec](#).

12.259.3.6 DecodeByStreams()

```
bool gdcmm::RAWCodec::DecodeByStreams (  
    std::istream & is,  
    std::ostream & os) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).

12.259.3.7 DecodeBytes()

```
bool gdcmm::RAWCodec::DecodeBytes (  
    const char * inBytes,  
    size_t inBufferLength,  
    char * outBytes,  
    size_t inOutBufferLength)
```

Used by the ImageStreamReader– converts a read in buffer into one with the proper encodings.

12.259.3.8 GetHeaderInfo()

```
bool gdcM::RAWCodec::GetHeaderInfo (  
    std::istream & is,  
    TransferSyntax & ts) [override], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

The documentation for this class was generated from the following file:

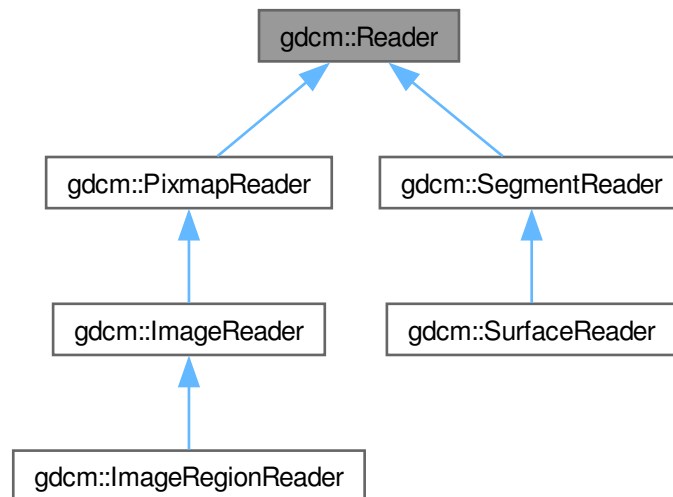
- [gdcMRAWCodec.h](#)

12.260 gdcM::Reader Class Reference

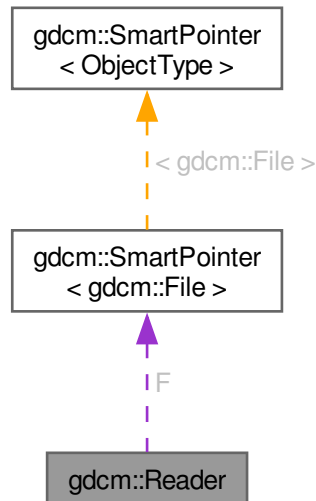
[Reader](#) ala DOM (Document [Object](#) Model).

```
#include <gdcMReader.h>
```

Inheritance diagram for gdcM::Reader:



Collaboration diagram for gdcmm::Reader:



Public Member Functions

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()
 - Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
 - Set/Get [File](#).
- size_t [GetStreamCurrentPosition](#) () const
- virtual bool [Read](#) ()
 - Main function to read a file.
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
 - Will only read the specified selected private tags.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
 - Will only read the specified selected tags.
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)
 - Set/Get [File](#).
- void [SetFileName](#) (const char *filename_native)
- void [SetStream](#) (std::istream &input_stream)
 - Set the open-ed stream directly.

Protected Member Functions

- `std::istream * GetStreamPtr () const`
- `bool ReadDataSet ()`
- `bool ReadMetaInformation ()`
- `bool ReadPreamble ()`

Protected Attributes

- `SmartPointer< File > F`

Friends

- `class StreamImageReader`

12.260.1 Detailed Description

[Reader](#) ala DOM (Document [Object](#) Model).

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailed description here

A [DataSet](#) DOES NOT contains group 0x0002 (see [FileMetaInformation](#))

This is really a [DataSet](#) reader. This will not make sure the dataset conform to any [IOD](#) at all. This is a completely different step. The reasoning was that user could control the [IOD](#) there lib would handle and thus we would not be able to read a [DataSet](#) if the [IOD](#) was not found Instead we separate the reading from the validation.

Note

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the [Value](#) to be read. By default it was 0xffff. The main author of GDCM2 thought this was too dangerous and harmful and therefore this feature did not make it into GDCM2

Warning

GDCM will not produce warning for unordered (non-alphabetical order).

See also

[Writer](#) [FileMetaInformation](#) [DataSet](#) [File](#)

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.260.2 Constructor & Destructor Documentation

12.260.2.1 Reader()

gdcm::Reader::Reader ()

12.260.2.2 ~Reader()

virtual gdcm::Reader::~Reader () [virtual]

12.260.3 Member Function Documentation

12.260.3.1 CanRead()

bool gdcm::Reader::CanRead () const

Test whether this is a DICOM file

Warning

need to call either SetFileName or SetStream first

Examples

[ReadUTF8QtDir.cxx](#).

12.260.3.2 GetFile() [1/2]

[File](#) & gdcm::Reader::GetFile () [inline]

Set/Get [File](#).

References [F](#).

12.260.3.3 GetFile() [2/2]

const [File](#) & gdcmm::Reader::GetFile () const [inline]

Set/Get [File](#).

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExplicitLittleEndian.cs](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StandardizeFiles.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtnionplan.cxx](#), [gdcmrtpplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

References [F](#).

12.260.3.4 GetStreamCurrentPosition()

size_t gdcmm::Reader::GetStreamCurrentPosition () const

For wrapped language. return type is compatible with [System::FileSize](#) return type Use native std::streampos / std::streamoff directly from the stream from C++

Examples

[ExtractImageRegion.cs](#).

12.260.3.5 GetStreamPtr()

std::istream * gdcmm::Reader::GetStreamPtr () const [inline], [protected]

12.260.3.6 Read()

virtual bool gdcm::Reader::Read () [virtual]

Main function to read a file.

Reimplemented in [gdcm::ImageReader](#), [gdcm::ImageRegionReader](#), [gdcm::PixmapReader](#), [gdcm::SegmentReader](#), and [gdcm::SurfaceReader](#).

Examples

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReformatFile.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.260.3.7 ReadDataSet()

bool gdcm::Reader::ReadDataSet () [protected]

12.260.3.8 ReadMetaInformation()

bool gdcm::Reader::ReadMetaInformation () [protected]

12.260.3.9 ReadPreamble()

bool gdcm::Reader::ReadPreamble () [protected]

12.260.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (  
    std::set< PrivateTag > const & ptags,  
    bool readvalues = true)
```

Will only read the specified selected private tags.

12.260.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
    std::set< Tag > const & tags,
    bool readvalues = true)
```

Will only read the specified selected tags.

12.260.3.12 ReadUpToTag()

```
bool gdcm::Reader::ReadUpToTag (
    const Tag & tag,
    std::set< Tag > const & skiptags = std::set< Tag >())
```

Will read only up to [Tag](#)

Parameters

tag	and skipping any tag specified in
skiptags	

Examples

[DumpVisusChange.cxx](#).

12.260.3.13 SetFile()

```
void gdcm::Reader::SetFile (
    File & file) [inline]
```

Set/Get [File](#).

References [F](#).

12.260.3.14 SetFileName()

```
void gdcm::Reader::SetFileName (
    const char * filename_native)
```

Set the filename to open. This will create a std::ifstream internally See SetStream if you are dealing with different std::istream object

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExplicitLittleEndian.cs](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetArray.cs](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [PrintLUT.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [ReformatFile.cs](#), [RescaleImage.cs](#), [SimplePrint.cs](#), [SimplePrintPatientName.cs](#), [StandardizeFiles.cs](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), [csa2img.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), [rle2img.cxx](#), and [threadgdcm.cxx](#).

12.260.3.15 SetStream()

```
void gdcm::Reader::SetStream (
    std::istream & input_stream) [inline]
```

Set the open-ed stream directly.

Examples

[ReadUTF8QtDir.cxx](#).

12.260.4 Friends And Related Symbol Documentation

12.260.4.1 StreamImageReader

```
friend class StreamImageReader [friend]
```

References [StreamImageReader](#).

Referenced by [StreamImageReader](#).

12.260.5 Member Data Documentation

12.260.5.1 F

```
SmartPointer<File> gdcm::Reader::F [protected]
```

Referenced by [GetFile\(\)](#), [GetFile\(\)](#), and [SetFile\(\)](#).

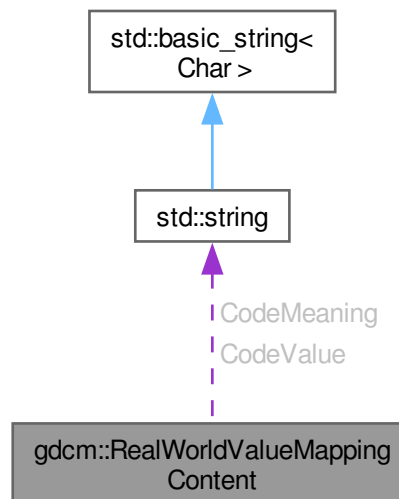
The documentation for this class was generated from the following file:

- [gdcmReader.h](#)

12.261 gdcmm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmmImageHelper.h>
```

Collaboration diagram for gdcmm::RealWorldValueMappingContent:



Public Attributes

- `std::string` [CodeMeaning](#)
- `std::string` [CodeValue](#)
- `double` [RealWorldValueIntercept](#)
- `double` [RealWorldValueSlope](#)

12.261.1 Member Data Documentation

12.261.1.1 CodeMeaning

`std::string gdcmm::RealWorldValueMappingContent::CodeMeaning`

12.261.1.2 CodeValue

`std::string gdcmm::RealWorldValueMappingContent::CodeValue`

12.261.1.3 RealWorldValueIntercept

double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept

12.261.1.4 RealWorldValueSlope

double gdcm::RealWorldValueMappingContent::RealWorldValueSlope

The documentation for this struct was generated from the following file:

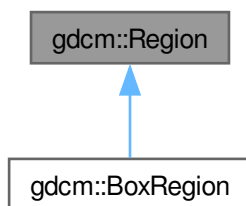
- [gdcmImageHelper.h](#)

12.262 gdcm::Region Class Reference

Class for manipulation region.

```
#include <gdcmRegion.h>
```

Inheritance diagram for gdcm::Region:



Public Member Functions

- [Region](#) ()
- virtual [~Region](#) ()
- virtual size_t [Area](#) () const =0
compute the area
- virtual [Region](#) * [Clone](#) () const =0
- virtual [BoxRegion](#) [ComputeBoundingBox](#) ()=0
Return the Axis-Aligned minimum bounding box for all regions.
- virtual bool [Empty](#) () const =0
return whether this domain is empty:
- virtual bool [IsValid](#) () const =0
return whether this is valid domain
- virtual void [Print](#) (std::ostream &os=std::cout) const
Print.

12.262.1 Detailed Description

Class for manipulation region.

12.262.2 Constructor & Destructor Documentation

12.262.2.1 Region()

`gdcM::Region::Region ()`

Referenced by [gdcM::BoxRegion::Clone\(\)](#), and [Clone\(\)](#).

12.262.2.2 ~Region()

`virtual gdcM::Region::~~Region () [virtual]`

12.262.3 Member Function Documentation

12.262.3.1 Area()

`virtual size_t gdcM::Region::Area () const [pure virtual]`

compute the area

Implemented in [gdcM::BoxRegion](#).

12.262.3.2 Clone()

`virtual Region * gdcM::Region::Clone () const [pure virtual]`

Implemented in [gdcM::BoxRegion](#).

References [Region\(\)](#).

12.262.3.3 ComputeBoundingBox()

`virtual BoxRegion gdcM::Region::ComputeBoundingBox () [pure virtual]`

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcM::BoxRegion](#).

12.262.3.4 Empty()

virtual bool gdcM::Region::Empty () const [pure virtual]

return whether this domain is empty:

Implemented in [gdcM::BoxRegion](#).

12.262.3.5 IsValid()

virtual bool gdcM::Region::IsValid () const [pure virtual]

return whether this is valid domain

Implemented in [gdcM::BoxRegion](#).

12.262.3.6 Print()

virtual void gdcM::Region::Print (
std::ostream & os = std::cout) const [virtual]

Print.

Reimplemented in [gdcM::BoxRegion](#).

Referenced by [gdcM::operator<<\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMRegion.h](#)

12.263 gdcM::Rescaler Class Reference

Rescale class.

```
#include <gdcMRescaler.h>
```

Public Member Functions

- [Rescaler](#) ()
- [~Rescaler](#) ()=default
- [PixelFormat::ScalarType ComputeInterceptSlopePixelFormat](#) ()
- [PixelFormat ComputePixelFormatFromMinMax](#) ()
- double [GetIntercept](#) () const
- double [GetSlope](#) () const
- bool [InverseRescale](#) (char *out, const char *in, size_t n)
Inverse transform.
- bool [Rescale](#) (char *out, const char *in, size_t n)
Direct transform.
- void [SetIntercept](#) (double i)
Set Intercept: used for both direct&inverse transformation.
- void [SetMinMaxForPixelFormat](#) (double min, double max)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
Set Pixel Format of input data.
- void [SetSlope](#) (double s)
Set Slope: user for both direct&inverse transformation.
- void [SetTargetPixelFormat](#) ([PixelFormat](#) const &targetst)
- void [SetUseTargetPixelFormat](#) (bool b)
Override default behavior of Rescale.

Protected Member Functions

- template<typename TIn>
void [InverseRescaleFunctionIntoBestFit](#) (char *out, const TIn *in, size_t n)
- template<typename TIn>
void [RescaleFunctionIntoBestFit](#) (char *out, const TIn *in, size_t n)

12.263.1 Detailed Description

Rescale class.

This class is meant to apply the linear transform of Stored Pixel [Value](#) to Real World [Value](#). This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel [Type](#) is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World [Value](#) will be 16 bits signed type.

In PET: the linear transform is generally floating point based. Since the dynamic range can be quite high, the Rescale Slope / Rescale Intercept can be changing throughout the [Series](#). So it is important to read all linear transform and deduce the best Pixel [Type](#) only at the end (when all the images to be read have been parsed).

Warning

Internally any time a floating point value is found either in the Rescale Slope or the Rescale Intercept it is assumed that the best matching output pixel type is FLOAT64 (in previous implementation it was FLOAT32). Because [VR:DS](#) is closer to a 64bits floating point type FLOAT64 is thus a best matching pixel type for the floating point transformation.

Example: Let say input is FLOAT64, and we want UINT16 as output, we would do:

```
Rescaler ir;
ir.SetIntercept( 0 );
ir.SetSlope( 5.6789 );
ir.SetPixelFormat( FLOAT64 );
ir.SetMinMaxForPixelType( ((PixelFormat)UINT16).GetMin(), ((PixelFormat)UINT16).GetMax() );
ir.InverseRescale(output,input,numberofbytes );
```

Note

handle floating point transformation back and forth to integer properly (no loss)

See also

[Unpacker12Bits](#)

Examples

[RescaleImage.cs.](#)

12.263.2 Constructor & Destructor Documentation

12.263.2.1 Rescaler()

gdcm::Rescaler::Rescaler () [inline]

12.263.2.2 ~Rescaler()

gdcm::Rescaler::~~Rescaler () [default]

12.263.3 Member Function Documentation

12.263.3.1 ComputeInterceptSlopePixelFormat()

[PixelFormat::ScalarType](#) gdcm::Rescaler::ComputeInterceptSlopePixelFormat ()

Compute the Pixel Format of the output data Used for direct transformation

Examples

[RescaleImage.cs.](#)

12.263.3.2 ComputePixelTypeFromMinMax()

[PixelFormat](#) gdcm::Rescaler::ComputePixelTypeFromMinMax ()

Compute the Pixel Format of the output data Used for inverse transformation

12.263.3.3 GetIntercept()

double gdcm::Rescaler::GetIntercept () const [inline]

12.263.3.4 GetSlope()

double gdcm::Rescaler::GetSlope () const [inline]

12.263.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
    char * out,
    const char * in,
    size_t n)
```

Inverse transform.

12.263.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn>
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n) [protected]
```

12.263.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
    char * out,
    const char * in,
    size_t n)
```

Direct transform.

Examples

[RescaleImage.cs](#).

12.263.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn>
void gdcmm::Rescaler::RescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n) [protected]
```

12.263.3.9 SetIntercept()

```
void gdcmm::Rescaler::SetIntercept (
    double i) [inline]
```

Set Intercept: used for both direct&inverse transformation.

Examples

[RescaleImage.cs](#).

12.263.3.10 SetMinMaxForPixelType()

```
void gdcmm::Rescaler::SetMinMaxForPixelType (
    double min,
    double max)
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

12.263.3.11 SetPixelFormat()

```
void gdcmm::Rescaler::SetPixelFormat (
    PixelFormat const & pf) [inline]
```

Set Pixel Format of input data.

Examples

[RescaleImage.cs](#).

12.263.3.12 SetSlope()

```
void gdcmm::Rescaler::SetSlope (
    double s) [inline]
```

Set Slope: user for both direct&inverse transformation.

Examples

[RescaleImage.cs](#).

12.263.3.13 SetTargetPixelFormat()

```
void gdcM::Rescaler::SetTargetPixelFormat (
    PixelFormat const & targetst)
```

By default (when UseTargetPixelFormat is false), a best matching Target Pixel [Type](#) is computed. However user can override this auto selection by switching UseTargetPixelFormat:true and also specifying the specific Target Pixel [Type](#)

12.263.3.14 SetUseTargetPixelFormat()

```
void gdcM::Rescaler::SetUseTargetPixelFormat (
    bool b)
```

Override default behavior of Rescale.

The documentation for this class was generated from the following file:

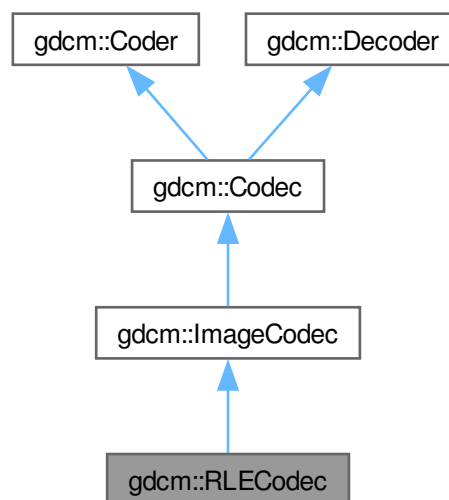
- [gdcMRescaler.h](#)

12.264 gdcM::RLECodec Class Reference

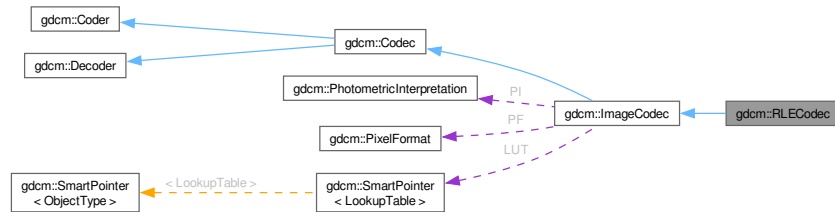
Class to do RLE.

```
#include <gdcMRLECodec.h>
```

Inheritance diagram for gdcM::RLECodec:



Collaboration diagram for gdcM::RLECodec:



Public Member Functions

- [RLECodec](#) ()
- [~RLECodec](#) () override
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const override
Return whether this coder support this transfer syntax (can code it).
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const override
Return whether this decoder support this transfer syntax (can decode it).
- [ImageCodec](#) * [Clone](#) () const override
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out) override
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os) override
Decode.
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts) override
- void [SetBufferLength](#) (unsigned long l)
- void [SetLength](#) (unsigned long l)

Public Member Functions inherited from [gdcM::ImageCodec](#)

- [ImageCodec](#) ()
- [~ImageCodec](#) () override
- bool [CleanupUnusedBits](#) (char *data, size_t datalen)
- const unsigned int * [GetDimensions](#) () const
- bool [GetLossyFlag](#) () const
- const [LookupTable](#) & [GetLUT](#) () const
- bool [GetNeedByteSwap](#) () const
- unsigned int [GetNumberOfDimensions](#) () const
- const [PhotometricInterpretation](#) & [GetPhotometricInterpretation](#) () const
- [PixelFormat](#) & [GetPixelFormat](#) ()
- const [PixelFormat](#) & [GetPixelFormat](#) () const
- unsigned int [GetPlanarConfiguration](#) () const
- bool [IsLossy](#) () const
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- void [SetDimensions](#) (const unsigned int d[3])

- void [SetLossyFlag](#) (bool l)
- void [SetLUT](#) ([LookupTable](#) const &lut)
- void [SetNeedByteSwap](#) (bool b)
- void [SetNeedOverlayCleanup](#) (bool b)
- void [SetNumberOfDimensions](#) (unsigned int dim)
- void [SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)
- virtual void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetPlanarConfiguration](#) (unsigned int pc)

Public Member Functions inherited from [gdcm::Coder](#)

- virtual [~Coder](#) ()=default

Public Member Functions inherited from [gdcm::Decoder](#)

- virtual [~Decoder](#) ()=default

Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [AppendRowEncode](#) (std::ostream &out, const char *data, size_t datalen) override
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os) override
- bool [DecodeExtent](#) (char *buffer, unsigned int XMin, unsigned int XMax, unsigned int YMin, unsigned int YMax, unsigned int ZMin, unsigned int ZMax, std::istream &is)
- bool [IsFrameEncoder](#) () override
- bool [IsRowEncoder](#) () override
- bool [StartEncode](#) (std::ostream &) override
- bool [StopEncode](#) (std::ostream &) override

Protected Member Functions inherited from [gdcm::ImageCodec](#)

- bool [DoByteSwap](#) (std::istream &is_, std::ostream &os)
- bool [DoInvertMonochrome](#) (std::istream &is_, std::ostream &os)
- bool [DoOverlayCleanup](#) (std::istream &is_, std::ostream &os)
- bool [DoPaddedCompositePixelCode](#) (std::istream &is_, std::ostream &os)
- bool [DoPlanarConfiguration](#) (std::istream &is_, std::ostream &os)
- bool [DoSimpleCopy](#) (std::istream &is_, std::ostream &os)
- bool [DoYBR](#) (std::istream &is_, std::ostream &os)
- bool [DoYBRFull422](#) (std::istream &is_, std::ostream &os)
- virtual bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)

Protected Member Functions inherited from [gdcm::Coder](#)

- virtual bool [InternalCode](#) (const char *bv, unsigned long len, std::ostream &os)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

Protected Types inherited from [gdcm::ImageCodec](#)

- typedef [SmartPointer](#)< [LookupTable](#) > [LUTPtr](#)

Protected Attributes inherited from [gdcm::ImageCodec](#)

- unsigned int [Dimensions](#) [3]
- bool [LossyFlag](#)
- [LUTPtr](#) [LUT](#)
- bool [NeedByteSwap](#)
- bool [NeedOverlayCleanup](#)
- unsigned int [NumberOfDimensions](#)
- [PixelFormat](#) [PF](#)
- [PhotometricInterpretation](#) [PI](#)
- unsigned int [PlanarConfiguration](#)
- bool [RequestPaddedCompositePixelCode](#)
- bool [RequestPlanarConfiguration](#)

12.264.1 Detailed Description

Class to do RLE.

Note

ANSI X3.9 A.4.2 RLE Compression Annex G defines a RLE Compression Transfer Syntax. This transfer Syntax is identified by the UID value "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one [Fragment](#) (see PS 3.5.8.2).

12.264.2 Constructor & Destructor Documentation

12.264.2.1 RLECodec()

gdcm::RLECodec::RLECodec ()

12.264.2.2 ~RLECodec()

gdcm::RLECodec::~~RLECodec () [override]

12.264.3 Member Function Documentation

12.264.3.1 AppendFrameEncode()

```
bool gdcmm::RLECodec::AppendFrameEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).

12.264.3.2 AppendRowEncode()

```
bool gdcmm::RLECodec::AppendRowEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen) [override], [protected], [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).

12.264.3.3 CanCode()

```
bool gdcmm::RLECodec::CanCode (  
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from [gdcmm::ImageCodec](#).

12.264.3.4 CanDecode()

```
bool gdcmm::RLECodec::CanDecode (  
    TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from [gdcmm::ImageCodec](#).

12.264.3.5 Clone()

```
ImageCodec * gdcmm::RLECodec::Clone () const [override], [virtual]
```

Implements [gdcmm::ImageCodec](#).

References [gdcmm::ImageCodec::ImageCodec\(\)](#).

12.264.3.6 Code()

```
bool gdcm::RLECodec::Code (
    DataElement const & in_,
    DataElement & out_) [override], [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

12.264.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
    DataElement const & ,
    DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.9 DecodeExtent()

```
bool gdcm::RLECodec::DecodeExtent (
    char * buffer,
    unsigned int XMin,
    unsigned int XMax,
    unsigned int YMin,
    unsigned int YMax,
    unsigned int ZMin,
    unsigned int ZMax,
    std::istream & is) [protected]
```

12.264.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength () const [inline]
```

12.264.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts) [override], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder () [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder () [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
    unsigned long l) [inline]
```

12.264.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
    unsigned long l) [inline]
```

12.264.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
    std::ostream & ) [override], [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

12.264.4 Friends And Related Symbol Documentation

12.264.4.1 ImageRegionReader

friend class ImageRegionReader [friend]

References [ImageRegionReader](#).

Referenced by [ImageRegionReader](#).

The documentation for this class was generated from the following file:

- [gdcmRLECodec.h](#)

12.265 gdcm::network::RoleSelectionSub Class Reference

[RoleSelectionSub](#).

```
#include <gdcmRoleSelectionSub.h>
```

Public Member Functions

- [RoleSelectionSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char *uid, uint8_t scurole, uint8_t scprole)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.265.1 Detailed Description

[RoleSelectionSub](#).

PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.265.2 Constructor & Destructor Documentation

12.265.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ()
```

12.265.3 Member Function Documentation

12.265.3.1 Print()

```
void gdcmm::network::RoleSelectionSub::Print (  
    std::ostream & os) const
```

12.265.3.2 Read()

```
std::istream & gdcmm::network::RoleSelectionSub::Read (  
    std::istream & is)
```

12.265.3.3 SetTuple()

```
void gdcmm::network::RoleSelectionSub::SetTuple (  
    const char * uid,  
    uint8_t scurole,  
    uint8_t scprole)
```

12.265.3.4 Size()

```
size_t gdcmm::network::RoleSelectionSub::Size () const
```

12.265.3.5 Write()

```
const std::ostream & gdcmm::network::RoleSelectionSub::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

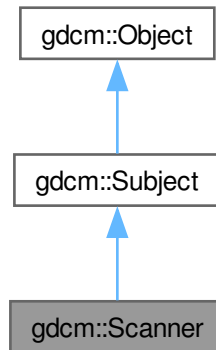
- [gdcmmRoleSelectionSub.h](#)

12.266 gdcm::Scanner Class Reference

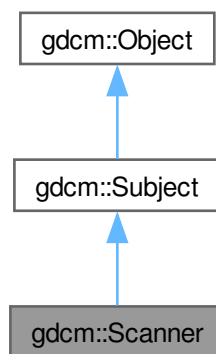
[Scanner](#).

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct [ltstr](#)

Public Types

- typedef MappingType::const_iterator [ConstIterator](#)
- typedef std::map< const char *, [TagToValue](#), ltstr > [MappingType](#)
- typedef std::map< [Tag](#), const char * > [TagToValue](#)
- typedef TagToValue::value_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) () override
- void [AddPrivateTag](#) ([PrivateTag](#) const &t)
- void [AddSkipTag](#) ([Tag](#) const &t)

Add a tag that will need to be skipped. Those are root level skip tags.
- void [AddTag](#) ([Tag](#) const &t)

Add a tag that will need to be read. Those are root level tags.
- [ConstIterator](#) [Begin](#) () const
- void [ClearSkipTags](#) ()
- void [ClearTags](#) ()
- [ConstIterator](#) [End](#) () const
- [Directory::FileNamesType](#) [GetAllFileNamesFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- const char * [GetFilenameFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- [Directory::FileNamesType](#) const & [GetFileNames](#) () const
- [Directory::FileNamesType](#) [GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char *filename) const

Get the std::map mapping filenames to value for file 'filename'.
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char *value) const

See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.
- [MappingType](#) const & [GetMappings](#) () const

Mappings are the mapping from a particular tag to the map, mapping filename to value:
- [Directory::FileNamesType](#) [GetOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetValue](#) (const char *filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const

Get all the values found (in lexicographic order).
- [ValuesType](#) [GetValues](#) ([Tag](#) const &t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.
- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const override

Print result.
- void [PrintTable](#) (std::ostream &os) const
- bool [Scan](#) ([Directory::FileNamesType](#) const &filenames)

Start the scan !

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer](#)< [Scanner](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Protected Member Functions

- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char *filename)

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Scanner](#) &s)

12.266.1 Detailed Description

Scanner.

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a `std::set of std::string`. Then the address of the `cstring` underlying the `std::string` is used in the `std::map`.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

Examples

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.266.2 Member Typedef Documentation

12.266.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcmm::Scanner::ConstIterator
```

12.266.2.2 MappingType

```
typedef std::map<const char *, TagToValue, ltstr> gdcmm::Scanner::MappingType
```

12.266.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcmm::Scanner::TagToValue
```

struct to map a filename to a value Implementation note: all `std::map` in this class will be using `const char *` and not `std::string` since we are pointing to existing `std::string` (hold in a `std::vector`) this avoid an extra copy of the byte array. [Tag](#) are used as [Tag](#) class since `sizeof(tag) <= sizeof(pointer)`

12.266.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

12.266.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

Examples

[DiscriminateVolume.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.266.3 Constructor & Destructor Documentation

12.266.3.1 Scanner()

```
gdcm::Scanner::Scanner () [inline]
```

Referenced by [New\(\)](#), and [operator<<](#).

12.266.3.2 ~Scanner()

```
gdcm::Scanner::~Scanner () [override]
```

12.266.4 Member Function Documentation

12.266.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (  
    PrivateTag const & t)
```

12.266.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (  
    Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.266.4.3 AddTag()

```
void gdcm::Scanner::AddTag (  
    Tag const & t)
```

Add a tag that will need to be read. Those are root level tags.

Examples

[DiscriminateVolume.cxx](#), [DumpToSQLite3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.266.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin () const [inline]
```

12.266.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ()
```

12.266.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ()
```

12.266.4.7 End()

```
ConstIterator gdcm::Scanner::End () const [inline]
```

12.266.4.8 GetAllFileNamesFromTagToValue()

```
Directory::FileNamesType gdcm::Scanner::GetAllFileNamesFromTagToValue (  
    Tag const & t,  
    const char * valuref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

12.266.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::Scanner::GetFilenameFromTagToValue (  
    Tag const & t,  
    const char * valuref) const
```

Will loop over all files and return the first file where value match the reference value 'valuref'

12.266.4.10 GetFilenames()

[Directory::FilenamesType](#) const & gdcm::Scanner::GetFilenames () const [inline]

12.266.4.11 GetKeys()

[Directory::FilenamesType](#) gdcm::Scanner::GetKeys () const

Return the list of filename that are key in the internal map, which means those filename were properly parsed

Examples

[VolumeSorter.cxx](#).

12.266.4.12 GetMapping()

[TagToValue](#) const & gdcm::Scanner::GetMapping (
const char * filename) const

Get the std::map mapping filenames to value for file 'filename'.

Examples

[DumpToSQLITE3.cxx](#).

12.266.4.13 GetMappingFromTagToValue()

[TagToValue](#) const & gdcm::Scanner::GetMappingFromTagToValue (
[Tag](#) const & t,
const char * value) const

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

12.266.4.14 GetMappings()

[MappingType](#) const & gdcm::Scanner::GetMappings () const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.266.4.15 GetOrderedValues()

[Directory::FilenamesType](#) gdcm::Scanner::GetOrderedValues (
[Tag](#) const & t) const

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.266.4.16 GetValue()

```
const char * gdcM::Scanner::GetValue (  
    const char * filename,  
    Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the [TagToValue](#) hash table.

Warning

[Tag 't'](#) should have been added via [AddTag\(\)](#) prior to the [Scan\(\)](#) call !

12.266.4.17 GetValues() [1/2]

```
ValueType const & gdcM::Scanner::GetValues () const [inline]
```

Get all the values found (in lexicographic order).

Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.266.4.18 GetValues() [2/2]

```
ValueType gdcM::Scanner::GetValues (  
    Tag const & t) const
```

Get all the values found (in lexicographic order) associated with [Tag 't'](#).

12.266.4.19 IsKey()

```
bool gdcM::Scanner::IsKey (  
    const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

Examples

[DumpToSQLITE3.cxx](#).

12.266.4.20 New()

[SmartPointer](#)< [Scanner](#) > gdcm::Scanner::New () [inline], [static]

for wrapped language: instantiate a reference counted object

References [Scanner\(\)](#).

12.266.4.21 Print()

```
void gdcm::Scanner::Print (
    std::ostream & os) const [override], [virtual]
```

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by [operator<<](#).

12.266.4.22 PrintTable()

```
void gdcm::Scanner::PrintTable (
    std::ostream & os) const
```

12.266.4.23 ProcessPublicTag()

```
void gdcm::Scanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename) [protected]
```

12.266.4.24 Scan()

```
bool gdcm::Scanner::Scan (
    Directory::FileNamesType const & filenames)
```

Start the scan !

Examples

[DiscriminateVolume.cxx](#), [DumpToSQLite3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.266.5 Friends And Related Symbol Documentation

12.266.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const Scanner & s) [friend]
```

References [Scanner\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

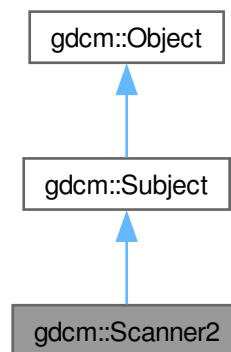
- [gdcmScanner.h](#)

12.267 gdcm::Scanner2 Class Reference

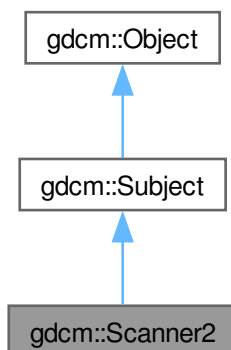
[Scanner2](#).

```
#include <gdcmScanner2.h>
```

Inheritance diagram for gdcm::Scanner2:



Collaboration diagram for gdcmm::Scanner2:



Classes

- struct [ltstr](#)

Public Types

- typedef PrivateMappingType::const_iterator [PrivateConstIterator](#)
- typedef std::map< const char *, [PrivateTagToValue](#), [ltstr](#) > [PrivateMappingType](#)
- typedef std::map< [PrivateTag](#), const char * > [PrivateTagToValue](#)
- typedef PrivateTagToValue::value_type [PrivateTagToValueValueType](#)
- typedef PublicMappingType::const_iterator [PublicConstIterator](#)
- typedef std::map< const char *, [PublicTagToValue](#), [ltstr](#) > [PublicMappingType](#)
- typedef std::map< [Tag](#), const char * > [PublicTagToValue](#)
- typedef PublicTagToValue::value_type [PublicTagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [Scanner2](#) ()
- [~Scanner2](#) () override
- bool [AddPrivateTag](#) ([PrivateTag](#) const &pt)
- bool [AddPublicTag](#) ([Tag](#) const &t)
Add a tag that will need to be read. Those are root level tags.
- bool [AddSkipTag](#) ([Tag](#) const &t)
Add a tag that will need to be skipped. Those are root level skip tags.
- [PublicConstIterator](#) [Begin](#) () const
- void [ClearPrivateTags](#) ()
- void [ClearPublicTags](#) ()

- void [ClearSkipTags](#) ()
- [PublicConstIterator End](#) () const
- [Directory::FileNamesType GetAllFileNamesFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *valueref) const
- [Directory::FileNamesType GetAllFileNamesFromPublicTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- const char * [GetFilenameFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *valueref) const
- const char * [GetFilenameFromPublicTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- [Directory::FileNamesType](#) const & [GetFileNames](#) () const

Return the list of filenames.

- [Directory::FileNamesType GetKeys](#) () const
- [PrivateTagToValue](#) const & [GetMappingFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *value) const
- [PublicTagToValue](#) const & [GetMappingFromPublicTagToValue](#) ([Tag](#) const &t, const char *value) const

See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.

- [PrivateTagToValue](#) const & [GetPrivateMapping](#) (const char *filename) const
- [PrivateMappingType](#) const & [GetPrivateMappings](#) () const
- [Directory::FileNamesType GetPrivateOrderedValues](#) ([PrivateTag](#) const &pt) const
- const char * [GetPrivateValue](#) (const char *filename, [PrivateTag](#) const &t) const
- [ValuesType](#) [GetPrivateValues](#) ([PrivateTag](#) const &pt) const

Get all the values found (in lexicographic order) associated with [PrivateTag](#) 'pt'.

- [PublicTagToValue](#) const & [GetPublicMapping](#) (const char *filename) const

Get the std::map mapping filenames to value for file 'filename'.

- [PublicMappingType](#) const & [GetPublicMappings](#) () const

Mappings are the mapping from a particular tag to the map, mapping filename to value:

- [Directory::FileNamesType GetPublicOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetPublicValue](#) (const char *filename, [Tag](#) const &t) const
- [ValuesType](#) [GetPublicValues](#) ([Tag](#) const &t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

- [ValuesType](#) const & [GetValues](#) () const

Get all the values found (in lexicographic order).

- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const override

Print result.

- void [PrintTable](#) (std::ostream &os, bool header=false) const

Print result as CSV table.

- [PrivateConstIterator PrivateBegin](#) () const
- [PrivateConstIterator PrivateEnd](#) () const
- bool [Scan](#) ([Directory::FileNamesType](#) const &filenames)

Start the scan !

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)

- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer](#)< [Scanner2](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Protected Member Functions

- void [ProcessPrivateTag](#) ([StringFilter](#) &sf, const char *filename)
- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char *filename)

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Scanner2](#) &s)

12.267.1 Detailed Description

[Scanner2](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a `std::set` of `std::string`. Then the address of the `cstring` underlying the `std::string` is used in the `std::map`.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

12.267.2 Member Typedef Documentation

12.267.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::Scanner2::PrivateConstIterator
```

12.267.2.2 PrivateMappingType

```
typedef std::map<const char *,PrivateTagToValue, ltstr> gdcm::Scanner2::PrivateMappingType
```

12.267.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char*> gdcm::Scanner2::PrivateTagToValue
```

12.267.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::Scanner2::PrivateTagToValueValueType
```

12.267.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::Scanner2::PublicConstIterator
```

12.267.2.6 PublicMappingType

```
typedef std::map<const char *,PublicTagToValue, ltstr> gdcm::Scanner2::PublicMappingType
```

12.267.2.7 PublicTagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

12.267.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::Scanner2::PublicTagToValueValueType
```

12.267.2.9 ValuesType

```
typedef std::set< std::string > gdcm::Scanner2::ValuesType
```

12.267.3 Constructor & Destructor Documentation

12.267.3.1 Scanner2()

```
gdcm::Scanner2::Scanner2 () [inline]
```

Referenced by [New\(\)](#), and [operator<<](#).

12.267.3.2 ~Scanner2()

```
gdcm::Scanner2::~~Scanner2 () [override]
```

12.267.4 Member Function Documentation

12.267.4.1 AddPrivateTag()

```
bool gdcm::Scanner2::AddPrivateTag (  
    PrivateTag const & pt)
```

12.267.4.2 AddPublicTag()

```
bool gdcm::Scanner2::AddPublicTag (  
    Tag const & t)
```

Add a tag that will need to be read. Those are root level tags.

12.267.4.3 AddSkipTag()

```
bool gdcmm::Scanner2::AddSkipTag (
    Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.267.4.4 Begin()

```
PublicConstIterator gdcmm::Scanner2::Begin () const [inline]
```

12.267.4.5 ClearPrivateTags()

```
void gdcmm::Scanner2::ClearPrivateTags ()
```

12.267.4.6 ClearPublicTags()

```
void gdcmm::Scanner2::ClearPublicTags ()
```

12.267.4.7 ClearSkipTags()

```
void gdcmm::Scanner2::ClearSkipTags ()
```

12.267.4.8 End()

```
PublicConstIterator gdcmm::Scanner2::End () const [inline]
```

12.267.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcmm::Scanner2::GetAllFilenamesFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valuref) const
```

12.267.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcmm::Scanner2::GetAllFilenamesFromPublicTagToValue (
    Tag const & t,
    const char * valuref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

12.267.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPrivateTagToValue (  
    PrivateTag const & pt,  
    const char * valueref) const
```

12.267.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPublicTagToValue (  
    Tag const & t,  
    const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.267.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner2::GetFilenames () const [inline]
```

Return the list of filenames.

12.267.4.14 GetKeys()

```
Directory::FilenamesType gdcm::Scanner2::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.267.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::Scanner2::GetMappingFromPrivateTagToValue (  
    PrivateTag const & pt,  
    const char * value) const
```

12.267.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::Scanner2::GetMappingFromPublicTagToValue (  
    Tag const & t,  
    const char * value) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

12.267.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::Scanner2::GetPrivateMapping (  
    const char * filename) const
```

12.267.4.18 GetPrivateMappings()

[PrivateMappingType](#) const & gdcmm::Scanner2::GetPrivateMappings () const [inline]

12.267.4.19 GetPrivateOrderedValues()

[Directory::FilenameType](#) gdcmm::Scanner2::GetPrivateOrderedValues (
[PrivateTag](#) const & pt) const

12.267.4.20 GetPrivateValue()

const char * gdcmm::Scanner2::GetPrivateValue (
 const char * filename,
[PrivateTag](#) const & t) const

12.267.4.21 GetPrivateValues()

[ValuesType](#) gdcmm::Scanner2::GetPrivateValues (
[PrivateTag](#) const & pt) const

Get all the values found (in lexicographic order) associated with [PrivateTag](#) 'pt'.

12.267.4.22 GetPublicMapping()

[PublicTagToValue](#) const & gdcmm::Scanner2::GetPublicMapping (
 const char * filename) const

Get the std::map mapping filenames to value for file 'filename'.

12.267.4.23 GetPublicMappings()

[PublicMappingType](#) const & gdcmm::Scanner2::GetPublicMappings () const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.267.4.24 GetPublicOrderedValues()

[Directory::FilenameType](#) gdcmm::Scanner2::GetPublicOrderedValues (
[Tag](#) const & t) const

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.267.4.25 GetPublicValue()

```
const char * gdcmm::Scanner2::GetPublicValue (  
    const char * filename,  
    Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.267.4.26 GetPublicValues()

```
ValuesType gdcmm::Scanner2::GetPublicValues (  
    Tag const & t) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

12.267.4.27 GetValues()

```
ValuesType const & gdcmm::Scanner2::GetValues () const [inline]
```

Get all the values found (in lexicographic order).

12.267.4.28 IsKey()

```
bool gdcmm::Scanner2::IsKey (  
    const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

12.267.4.29 New()

```
SmartPointer< Scanner2 > gdcmm::Scanner2::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References Scanner2().

12.267.4.30 Print()

```
void gdcm::Scanner2::Print (  
    std::ostream & os) const    [override], [virtual]
```

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by [operator<<](#).

12.267.4.31 PrintTable()

```
void gdcm::Scanner2::PrintTable (  
    std::ostream & os,  
    bool header = false) const
```

Print result as CSV table.

12.267.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::Scanner2::PrivateBegin () const    [inline]
```

12.267.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::Scanner2::PrivateEnd () const    [inline]
```

12.267.4.34 ProcessPrivateTag()

```
void gdcm::Scanner2::ProcessPrivateTag (  
    StringFilter & sf,  
    const char * filename)    [protected]
```

12.267.4.35 ProcessPublicTag()

```
void gdcm::Scanner2::ProcessPublicTag (  
    StringFilter & sf,  
    const char * filename)    [protected]
```

12.267.4.36 Scan()

```
bool gdcm::Scanner2::Scan (  
    Directory::FileNamesType const & filenames)
```

Start the scan !

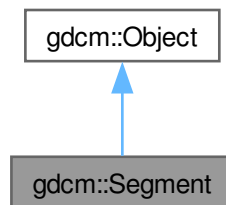
12.267.5.1 operator<<

References [Scanner2\(\)](#), and [Print\(\)](#).

- [gdcmScanner2.h](#)

This class defines a segment.

Inheritance diagram for `gdcm::Segment`:



```

graph TD
    gdcm_Object[gdcm::Object]
    gdcm_Segment[gdcm::Segment]
    gdcm_SegmentHelper[gdcm::SegmentHelper::BasicCodedEntry]
    std_string[std::string]
    std_vector_T[std::vector<T>]
    std_vector_Surfaces[std::vector<SmartPointer<Surface>>]
    T[T]

    gdcm_Object -- "SegmentAlgorithmName  
SegmentDescription  
SegmentLabel" --> gdcm_SegmentHelper
    gdcm_Segment -- "AnatomicRegion  
PropertyCategory  
PropertyType  
AnatomicRegionModifiers  
PropertyTypeModifiers" --> gdcm_SegmentHelper
    gdcm_SegmentHelper -- "Surfaces" --> std_vector_Surfaces
    std_vector_Surfaces -- "elements" --> T
    std_vector_T -- "elements" --> T
    std_string --> gdcm_SegmentHelper
    std_vector_T --> gdcm_SegmentHelper
    std_vector_Surfaces --> gdcm_SegmentHelper
    gdcm_SegmentHelper -- "CM  
CSD  
CSV  
CV" --> std_string
    gdcm_SegmentHelper -- "<SegmentHelper::BasicCodedEntry>" --> std_string
    gdcm_SegmentHelper -- "<SmartPointer<Surface>>" --> std_vector_Surfaces
  
```

Public Types

- enum [ALGOType](#) {
[AUTOMATIC](#) = 0 ,
[SEMIAUTOMATIC](#) ,
[MANUAL](#) ,
[ALGOType_END](#) }
- typedef std::vector< [SegmentHelper::BasicCodedEntry](#) > [BasicCodedEntryVector](#)
- typedef std::vector< [SmartPointer](#)< [Surface](#) > > [SurfaceVector](#)

Public Member Functions

- [Segment](#) ()
- [~Segment](#) () override
- void [AddSurface](#) ([SmartPointer](#)< [Surface](#) > surface)
- [SegmentHelper::BasicCodedEntry](#) & [GetAnatomicRegion](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetAnatomicRegion](#) () const
- [BasicCodedEntryVector](#) & [GetAnatomicRegionModifiers](#) ()
- [BasicCodedEntryVector](#) const & [GetAnatomicRegionModifiers](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyCategory](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyCategory](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyType](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyType](#) () const
- [BasicCodedEntryVector](#) & [GetPropertyTypeModifiers](#) ()
- [BasicCodedEntryVector](#) const & [GetPropertyTypeModifiers](#) () const
- const char * [GetSegmentAlgorithmName](#) () const
- [ALGOType](#) [GetSegmentAlgorithmType](#) () const
- const char * [GetSegmentDescription](#) () const
- const char * [GetSegmentLabel](#) () const
- unsigned short [GetSegmentNumber](#) () const
- [SmartPointer](#)< [Surface](#) > [GetSurface](#) (const unsigned int idx=0) const
- unsigned long [GetSurfaceCount](#) ()
- [SurfaceVector](#) & [GetSurfaces](#) ()
- [SurfaceVector](#) const & [GetSurfaces](#) () const
- void [SetAnatomicRegion](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetAnatomicRegionModifiers](#) ([BasicCodedEntryVector](#) const &BSEV)
- void [SetPropertyCategory](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyType](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyTypeModifiers](#) ([BasicCodedEntryVector](#) const &BSEV)
- void [SetSegmentAlgorithmName](#) (const char *name)
- void [SetSegmentAlgorithmType](#) ([ALGOType](#) type)
- void [SetSegmentAlgorithmType](#) (const char *typeStr)
- void [SetSegmentDescription](#) (const char *description)
- void [SetSegmentLabel](#) (const char *label)
- void [SetSegmentNumber](#) (const unsigned short num)
- void [SetSurfaceCount](#) (const unsigned long nb)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [ALGOType](#) [GetALGOType](#) (const char *type)
- static const char * [GetALGOTypeString](#) ([ALGOType](#) type)

Protected Attributes

- [SegmentHelper::BasicCodedEntry](#) [AnatomicRegion](#)
- [BasicCodedEntryVector](#) [AnatomicRegionModifiers](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyCategory](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyType](#)
- [BasicCodedEntryVector](#) [PropertyTypeModifiers](#)
- std::string [SegmentAlgorithmName](#)
- [ALGOType](#) [SegmentAlgorithmType](#)
- std::string [SegmentDescription](#)
- std::string [SegmentLabel](#)
- unsigned short [SegmentNumber](#)
- unsigned long [SurfaceCount](#)
- [SurfaceVector](#) [Surfaces](#)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.268.1 Detailed Description

This class defines a segment.

It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

See also

PS 3.3 C.8.20.2 and C.8.23

12.268.2 Member Typedef Documentation

12.268.2.1 BasicCodedEntryVector

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcmm::Segment::BasicCodedEntryVector
```

12.268.2.2 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcmm::Segment::SurfaceVector
```

12.268.3 Member Enumeration Documentation

12.268.3.1 ALGOType

```
enum gdcmm::Segment::ALGOType
```

Enumerator

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

12.268.4 Constructor & Destructor Documentation

12.268.4.1 Segment()

```
gdcmm::Segment::Segment ()
```

12.268.4.2 ~Segment()

```
gdcmm::Segment::~~Segment () \[override\]
```

12.268.5 Member Function Documentation

12.268.5.1 AddSurface()

```
void gdcmm::Segment::AddSurface (  
    SmartPointer< Surface > surface)
```

References [gdcmm::Object::SmartPointer](#).

12.268.5.2 GetALGOType()

[ALGOType](#) gdcm::Segment::GetALGOType (
const char * type) [static]

12.268.5.3 GetALGOTypeString()

const char * gdcm::Segment::GetALGOTypeString (
[ALGOType](#) type) [static]

12.268.5.4 GetAnatomicRegion() [1/2]

[SegmentHelper::BasicCodedEntry](#) & gdcm::Segment::GetAnatomicRegion ()

12.268.5.5 GetAnatomicRegion() [2/2]

[SegmentHelper::BasicCodedEntry](#) const & gdcm::Segment::GetAnatomicRegion () const

12.268.5.6 GetAnatomicRegionModifiers() [1/2]

[BasicCodedEntryVector](#) & gdcm::Segment::GetAnatomicRegionModifiers ()

12.268.5.7 GetAnatomicRegionModifiers() [2/2]

[BasicCodedEntryVector](#) const & gdcm::Segment::GetAnatomicRegionModifiers () const

12.268.5.8 GetPropertyCategory() [1/2]

[SegmentHelper::BasicCodedEntry](#) & gdcm::Segment::GetPropertyCategory ()

12.268.5.9 GetPropertyCategory() [2/2]

[SegmentHelper::BasicCodedEntry](#) const & gdcm::Segment::GetPropertyCategory () const

12.268.5.10 GetPropertyType() [1/2]

[SegmentHelper::BasicCodedEntry](#) & gdcm::Segment::GetPropertyType ()

12.268.5.11 `GetPropertyType()` [2/2]

[SegmentHelper::BasicCodedEntry](#) const & `gdcmm::Segment::GetPropertyType ()` const

12.268.5.12 `GetPropertyTypeModifiers()` [1/2]

[BasicCodedEntryVector](#) & `gdcmm::Segment::GetPropertyTypeModifiers ()`

12.268.5.13 `GetPropertyTypeModifiers()` [2/2]

[BasicCodedEntryVector](#) const & `gdcmm::Segment::GetPropertyTypeModifiers ()` const

12.268.5.14 `GetSegmentAlgorithmName()`

const char * `gdcmm::Segment::GetSegmentAlgorithmName ()` const

12.268.5.15 `GetSegmentAlgorithmType()`

[ALGOType](#) `gdcmm::Segment::GetSegmentAlgorithmType ()` const

12.268.5.16 `GetSegmentDescription()`

const char * `gdcmm::Segment::GetSegmentDescription ()` const

12.268.5.17 `GetSegmentLabel()`

const char * `gdcmm::Segment::GetSegmentLabel ()` const

12.268.5.18 `GetSegmentNumber()`

unsigned short `gdcmm::Segment::GetSegmentNumber ()` const

12.268.5.19 `GetSurface()`

[SmartPointer< Surface >](#) `gdcmm::Segment::GetSurface (`
 const unsigned int idx = 0) const

References [gdcmm::Object::SmartPointer](#).

12.268.5.20 GetSurfaceCount()

unsigned long gdcm::Segment::GetSurfaceCount ()

12.268.5.21 GetSurfaces() [1/2]

[SurfaceVector](#) & gdcm::Segment::GetSurfaces ()

12.268.5.22 GetSurfaces() [2/2]

[SurfaceVector](#) const & gdcm::Segment::GetSurfaces () const

12.268.5.23 SetAnatomicRegion()

void gdcm::Segment::SetAnatomicRegion (
 [SegmentHelper::BasicCodedEntry](#) const & BSE)

12.268.5.24 SetAnatomicRegionModifiers()

void gdcm::Segment::SetAnatomicRegionModifiers (
 [BasicCodedEntryVector](#) const & BSEV)

12.268.5.25 SetPropertyCategory()

void gdcm::Segment::SetPropertyCategory (
 [SegmentHelper::BasicCodedEntry](#) const & BSE)

12.268.5.26 SetPropertyType()

void gdcm::Segment::SetPropertyType (
 [SegmentHelper::BasicCodedEntry](#) const & BSE)

12.268.5.27 SetPropertyTypeModifiers()

void gdcm::Segment::SetPropertyTypeModifiers (
 [BasicCodedEntryVector](#) const & BSEV)

12.268.5.28 SetSegmentAlgorithmName()

void gdcm::Segment::SetSegmentAlgorithmName (
 const char * name)

12.268.5.29 SetSegmentAlgorithmType() [1/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (  
    ALGOType type)
```

12.268.5.30 SetSegmentAlgorithmType() [2/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (  
    const char * typeStr)
```

12.268.5.31 SetSegmentDescription()

```
void gdcm::Segment::SetSegmentDescription (  
    const char * description)
```

12.268.5.32 SetSegmentLabel()

```
void gdcm::Segment::SetSegmentLabel (  
    const char * label)
```

12.268.5.33 SetSegmentNumber()

```
void gdcm::Segment::SetSegmentNumber (  
    const unsigned short num)
```

12.268.5.34 SetSurfaceCount()

```
void gdcm::Segment::SetSurfaceCount (  
    const unsigned long nb)
```

12.268.6 Member Data Documentation

12.268.6.1 AnatomicRegion

[SegmentHelper::BasicCodedEntry](#) gdcm::Segment::AnatomicRegion [protected]

12.268.6.2 AnatomicRegionModifiers

[BasicCodedEntryVector](#) gdcm::Segment::AnatomicRegionModifiers [protected]

12.268.6.3 PropertyCategory

[SegmentHelper::BasicCodedEntry](#) gdcm::Segment::PropertyCategory [protected]

12.268.6.4 PropertyType

[SegmentHelper::BasicCodedEntry](#) gdcm::Segment::PropertyType [protected]

12.268.6.5 PropertyTypeModifiers

[BasicCodedEntryVector](#) gdcm::Segment::PropertyTypeModifiers [protected]

12.268.6.6 SegmentAlgorithmName

std::string gdcm::Segment::SegmentAlgorithmName [protected]

12.268.6.7 SegmentAlgorithmType

[ALGOType](#) gdcm::Segment::SegmentAlgorithmType [protected]

12.268.6.8 SegmentDescription

std::string gdcm::Segment::SegmentDescription [protected]

12.268.6.9 SegmentLabel

std::string gdcm::Segment::SegmentLabel [protected]

12.268.6.10 SegmentNumber

unsigned short gdcm::Segment::SegmentNumber [protected]

12.268.6.11 SurfaceCount

unsigned long gdcm::Segment::SurfaceCount [protected]

12.268.6.12 Surfaces

[SurfaceVector](#) gdcM::Segment::Surfaces [protected]

The documentation for this class was generated from the following file:

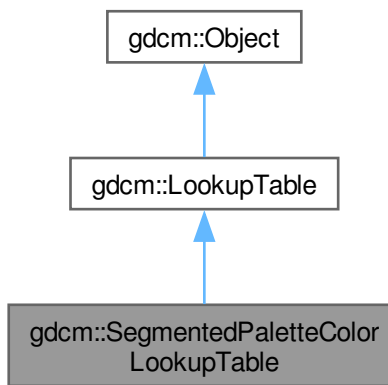
- [gdcMSegment.h](#)

12.269 gdcM::SegmentedPaletteColorLookupTable Class Reference

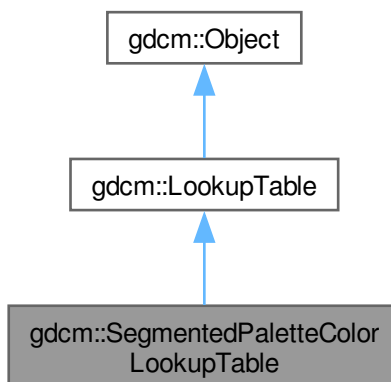
[SegmentedPaletteColorLookupTable](#) class.

```
#include <gdcMSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcM::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- [SegmentedPaletteColorLookupTable](#) ()
- [~SegmentedPaletteColorLookupTable](#) () override
- void [Print](#) (std::ostream &) const override
- void [SetLUT](#) ([LookupTableType](#) type, const unsigned char *array, unsigned int length) override
Initialize a [SegmentedPaletteColorLookupTable](#).

Public Member Functions inherited from [gdcm::LookupTable](#)

- [LookupTable](#) ()
- [LookupTable](#) (LookupTable const &lut)
- [~LookupTable](#) () override
- void [Allocate](#) (unsigned short bitsample=8)
Allocate the LUT.
- void [Clear](#) ()
Clear the LUT.
- bool [Decode](#) (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- void [Decode](#) (std::istream &is, std::ostream &os) const
Decode the LUT.
- bool [Decode8](#) (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
Decode into RGB 8 bits space.
- unsigned short [GetBitSample](#) () const
return the bit sample
- bool [GetBufferAsRGBA](#) (unsigned char *rgba) const
return the LUT as RGBA buffer
- void [GetLUT](#) ([LookupTableType](#) type, unsigned char *array, unsigned int &length) const

- void [GetLUTDescriptor](#) ([LookupTableType](#) type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int [GetLUTLength](#) ([LookupTableType](#) type) const
- const unsigned char * [GetPointer](#) () const
return a raw pointer to the LUT
- void [InitializeBlueLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool [Initialized](#) () const
return whether the LUT has been initialized
- void [InitializeGreenLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void [InitializeLUT](#) ([LookupTableType](#) type, unsigned short length, unsigned short subscript, unsigned short bitsize)
- Generic interface:
- void [InitializeRedLUT](#) (unsigned short length, unsigned short subscript, unsigned short bitsize)
RED / GREEN / BLUE specific:
- bool [IsRGB8](#) () const
Return whether 16 bits LUT is in RGB 8 bits space.
- void [SetBlueLUT](#) (const unsigned char *blue, unsigned int length)
- void [SetGreenLUT](#) (const unsigned char *green, unsigned int length)
- void [SetRedLUT](#) (const unsigned char *red, unsigned int length)
- bool [WriteBufferAsRGBA](#) (const unsigned char *rgba)
Write the LUT as RGBA.

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Additional Inherited Members

Public Types inherited from [gdcm::LookupTable](#)

- enum [LookupTableType](#) {
 [RED](#) = 0 ,
 [GREEN](#) ,
 [BLUE](#) ,
 [GRAY](#) ,
 [UNKNOWN](#) }

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes inherited from [gdcm::LookupTable](#)

- unsigned short [BitSample](#)
- bool [IncompleteLUT:1](#)
- LookupTableInternal * [Internal](#)

12.269.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

12.269.2 Constructor & Destructor Documentation

12.269.2.1 SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ()
```

12.269.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable () [override]
```

12.269.3 Member Function Documentation

12.269.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (  
    std::ostream & ) const [inline], [override], [virtual]
```

Reimplemented from [gdcm::LookupTable](#).

12.269.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (  
    LookupTableType type,  
    const unsigned char * array,  
    unsigned int length) [override], [virtual]
```

Initialize a [SegmentedPaletteColorLookupTable](#).

Reimplemented from [gdcm::LookupTable](#).

The documentation for this class was generated from the following file:

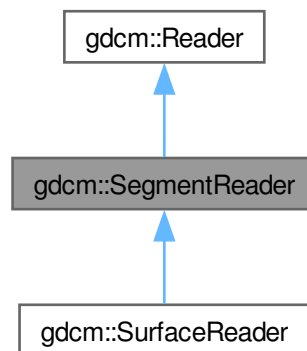
- [gdcmSegmentedPaletteColorLookupTable.h](#)

12.270 gdcM::SegmentReader Class Reference

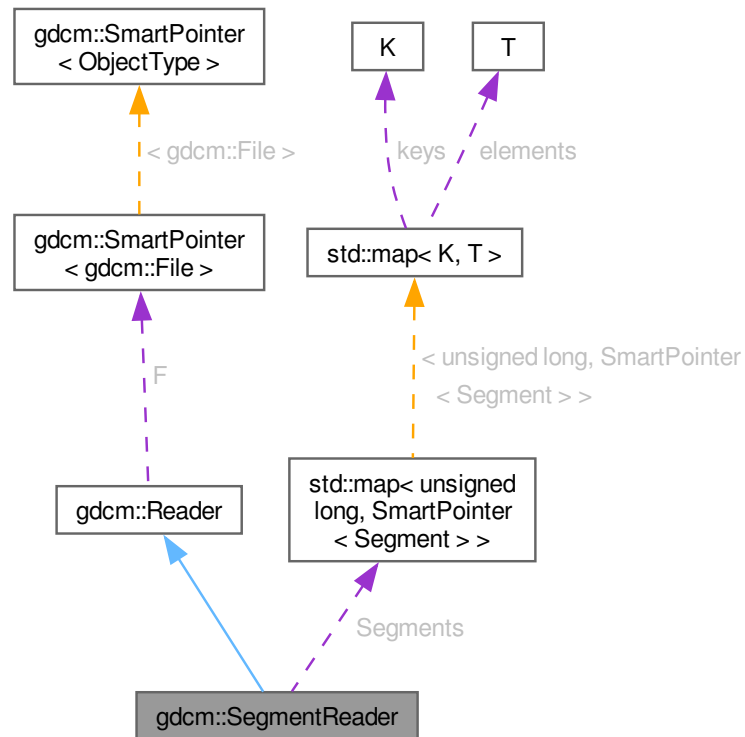
This class defines a segment reader.

```
#include <gdcMSegmentReader.h>
```

Inheritance diagram for gdcM::SegmentReader:



Collaboration diagram for gdcm::SegmentReader:



Public Types

- typedef std::vector< [SmartPointer< Segment >](#) > [SegmentVector](#)

Public Member Functions

- [SegmentReader](#) ()
- [~SegmentReader](#) () override
- [SegmentVector](#) [GetSegments](#) ()
- [SegmentVector](#) [GetSegments](#) () const
- bool [Read](#) () override

Read.

Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()
Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
Set/Get [File](#).
- size_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
Will only read the specified selected private tags.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
Will only read the specified selected tags.
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)
Set/Get [File](#).
- void [SetFileName](#) (const char *filename_native)
- void [SetStream](#) (std::istream &input_stream)
Set the open-ed stream directly.

Protected Types

- typedef std::map< unsigned long, [SmartPointer](#)< [Segment](#) > > [SegmentMap](#)

Protected Member Functions

- bool [ReadSegment](#) (const [Item](#) &segmentItem, const unsigned int idx)
- bool [ReadSegments](#) ()

Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream * [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

Protected Attributes

- [SegmentMap](#) [Segments](#)

Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer](#)< [File](#) > [F](#)

12.270.1 Detailed Description

This class defines a segment reader.

It reads attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

12.270.2 Member Typedef Documentation

12.270.2.1 SegmentMap

`typedef std::map< unsigned long, SmartPointer< Segment > > gdcmm::SegmentReader::SegmentMap [protected]`

12.270.2.2 SegmentVector

`typedef std::vector< SmartPointer< Segment > > gdcmm::SegmentReader::SegmentVector`

12.270.3 Constructor & Destructor Documentation

12.270.3.1 SegmentReader()

`gdcmm::SegmentReader::SegmentReader ()`

12.270.3.2 ~SegmentReader()

`gdcmm::SegmentReader::~~SegmentReader () [override]`

12.270.4 Member Function Documentation

12.270.4.1 GetSegments() [1/2]

`SegmentVector gdcmm::SegmentReader::GetSegments ()`

12.270.4.2 GetSegments() [2/2]

`SegmentVector gdcmm::SegmentReader::GetSegments () const`

12.270.4.3 Read()

bool gdcm::SegmentReader::Read () [override], [virtual]

Read.

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::SurfaceReader](#).

12.270.4.4 ReadSegment()

bool gdcm::SegmentReader::ReadSegment (
 const [Item](#) & segmentItem,
 const unsigned int idx) [protected]

12.270.4.5 ReadSegments()

bool gdcm::SegmentReader::ReadSegments () [protected]

12.270.5 Member Data Documentation

12.270.5.1 Segments

[SegmentMap](#) gdcm::SegmentReader::Segments [protected]

The documentation for this class was generated from the following file:

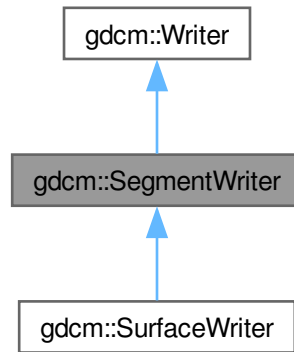
- [gdcmSegmentReader.h](#)

12.271 gdcm::SegmentWriter Class Reference

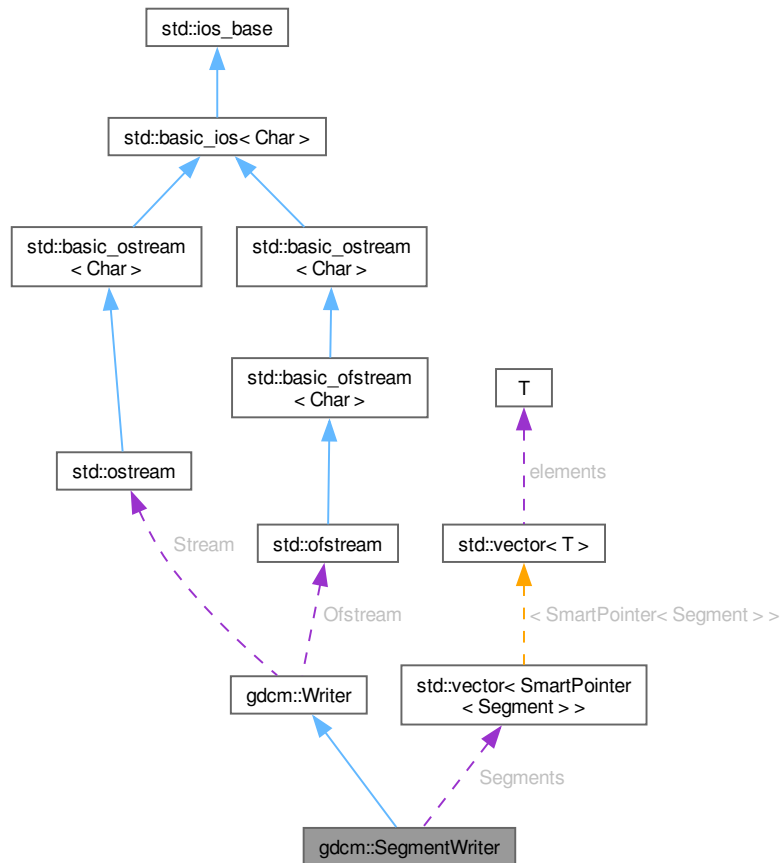
This class defines a segment writer.

```
#include <gdcmSegmentWriter.h>
```

Inheritance diagram for gdcmm::SegmentWriter:



Collaboration diagram for `gdcM::SegmentWriter`:



Public Types

- typedef `std::vector< SmartPointer< Segment > >` `SegmentVector`

Public Member Functions

- `SegmentWriter` ()
- `~SegmentWriter` () override
- void `AddSegment` (`SmartPointer< Segment >` segment)
- unsigned int `GetNumberOfSegments` () const
- `SmartPointer< Segment >` `GetSegment` (const unsigned int idx=0) const
- `SegmentVector` & `GetSegments` ()
- const `SegmentVector` & `GetSegments` () const
- void `SetNumberOfSegments` (const unsigned int size)
- void `SetSegments` (`SegmentVector` &segments)
- bool `Write` () override

Write.

Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)
Undocumented function, do not use (= leave default).
- void [SetFile](#) (const [File](#) &f)
Set/Get the DICOM file ([DataSet](#) + Header).
- void [SetFileName](#) (const char *filename_native)
Set the filename of DICOM file to write:
- void [SetStream](#) (std::ostream &output_stream)
Set user ostream buffer.

Protected Member Functions

- bool [PrepareWrite](#) ()

Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes

- [SegmentVector](#) [Segments](#)

Protected Attributes inherited from [gdcm::Writer](#)

- std::ofstream * [Ofstream](#)
- std::ostream * [Stream](#)

12.271.1 Detailed Description

This class defines a segment writer.

It writes attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

12.271.2 Member Typedef Documentation

12.271.2.1 SegmentVector

`typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector`

12.271.3 Constructor & Destructor Documentation

12.271.3.1 SegmentWriter()

`gdcm::SegmentWriter::SegmentWriter ()`

12.271.3.2 ~SegmentWriter()

`gdcm::SegmentWriter::~~SegmentWriter () [override]`

12.271.4 Member Function Documentation

12.271.4.1 AddSegment()

`void gdcm::SegmentWriter::AddSegment (
 SmartPointer< Segment > segment)`

12.271.4.2 GetNumberOfSegments()

`unsigned int gdcm::SegmentWriter::GetNumberOfSegments () const`

12.271.4.3 GetSegment()

`SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
 const unsigned int idx = 0) const`

12.271.4.4 GetSegments() [1/2]

`SegmentVector & gdcm::SegmentWriter::GetSegments ()`

12.271.4.5 GetSegments() [2/2]

`const SegmentVector & gdcm::SegmentWriter::GetSegments () const`

12.271.4.6 PrepareWrite()

bool gdcm::SegmentWriter::PrepareWrite () [protected]

12.271.4.7 SetNumberOfSegments()

void gdcm::SegmentWriter::SetNumberOfSegments (
 const unsigned int size)

12.271.4.8 SetSegments()

void gdcm::SegmentWriter::SetSegments (
 [SegmentVector](#) & segments)

12.271.4.9 Write()

bool gdcm::SegmentWriter::Write () [override], [virtual]

Write.

Reimplemented from [gdcm::Writer](#).

Reimplemented in [gdcm::SurfaceWriter](#).

12.271.5 Member Data Documentation

12.271.5.1 Segments

[SegmentVector](#) gdcm::SegmentWriter::Segments [protected]

The documentation for this class was generated from the following file:

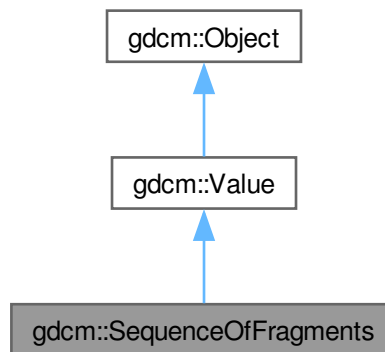
- [gdcmSegmentWriter.h](#)

12.272 gdcM::SequenceOfFragments Class Reference

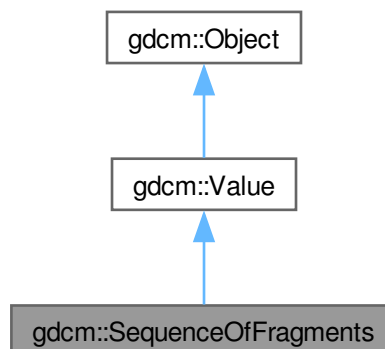
Class to represent a Sequence Of Fragments.

```
#include <gdcMSequenceOfFragments.h>
```

Inheritance diagram for gdcM::SequenceOfFragments:



Collaboration diagram for gdcM::SequenceOfFragments:



Public Types

- typedef `FragmentVector::const_iterator` [ConstIterator](#)
- typedef `std::vector< Fragment >` [FragmentVector](#)
- typedef `FragmentVector::iterator` [Iterator](#)
- typedef `FragmentVector::size_type` [SizeType](#)

Public Member Functions

- [SequenceOfFragments](#) ()
constructor (UndefinedLength by default)
- void [AddFragment](#) ([Fragment](#) const &item)
Appends a [Fragment](#) to the already added ones.
- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) () override
Clear.
- unsigned long [ComputeByteLength](#) () const
- [VL ComputeLength](#) () const
- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [GetBuffer](#) (char *buffer, unsigned long length) const
- bool [GetFragBuffer](#) (unsigned int fragNb, char *buffer, unsigned long &length) const
- const [Fragment](#) & [GetFragment](#) ([SizeType](#) num) const
- [VL GetLength](#) () const override
Returns the SQ length, as read from disk.
- [SizeType GetNumberOfFragments](#) () const
- [BasicOffsetTable](#) & [GetTable](#) ()
- const [BasicOffsetTable](#) & [GetTable](#) () const
- bool [operator==](#) (const [Value](#) &val) const override
- void [Print](#) (std::ostream &os) const override
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool)
- void [SetLength](#) ([VL](#) length) override
Sets the actual SQ length.
- template<typename TSwap>
std::ostream const & [Write](#) (std::ostream &os) const
- bool [WriteBuffer](#) (std::ostream &os) const

Public Member Functions inherited from [gdcmm::Value](#)

- [Value](#) ()=default
- [~Value](#) () override=default

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer< SequenceOfFragments > New](#) ()

Additional Inherited Members

Protected Member Functions inherited from [gdcM::Value](#)

- virtual void [SetLengthOnly](#) (VL l)

Protected Member Functions inherited from [gdcM::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.272.1 Detailed Description

Class to represent a Sequence Of Fragments.

[Todo](#) I do not enforce that Sequence of Fragments ends with a SQ end del

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), and [MpegVideoInfo.cs](#).

12.272.2 Member Typedef Documentation

12.272.2.1 ConstIterator

typedef [FragmentVector::const_iterator](#) [gdcM::SequenceOfFragments::ConstIterator](#)

12.272.2.2 FragmentVector

typedef [std::vector<Fragment>](#) [gdcM::SequenceOfFragments::FragmentVector](#)

12.272.2.3 Iterator

typedef [FragmentVector::iterator](#) [gdcM::SequenceOfFragments::Iterator](#)

12.272.2.4 SizeType

typedef FragmentVector::size_type [gdcm::SequenceOfFragments::SizeType](#)

12.272.3 Constructor & Destructor Documentation

12.272.3.1 SequenceOfFragments()

[gdcm::SequenceOfFragments::SequenceOfFragments](#) () [inline]

constructor (UndefinedLength by default)

Referenced by [New\(\)](#), and [operator==\(\)](#).

12.272.4 Member Function Documentation

12.272.4.1 AddFragment()

void [gdcm::SequenceOfFragments::AddFragment](#) (
 [Fragment](#) const & item)

Appends a [Fragment](#) to the already added ones.

12.272.4.2 Begin() [1/2]

[Iterator](#) [gdcm::SequenceOfFragments::Begin](#) () [inline]

Referenced by [Print\(\)](#), and [Write\(\)](#).

12.272.4.3 Begin() [2/2]

[ConstIterator](#) [gdcm::SequenceOfFragments::Begin](#) () const [inline]

12.272.4.4 Clear()

void [gdcm::SequenceOfFragments::Clear](#) () [override], [virtual]

Clear.

Implements [gdcm::Value](#).

12.272.4.5 ComputeByteLength()

unsigned long gdcm::SequenceOfFragments::ComputeByteLength () const

12.272.4.6 ComputeLength()

[VL](#) gdcm::SequenceOfFragments::ComputeLength () const

12.272.4.7 End() [1/2]

[Iterator](#) gdcm::SequenceOfFragments::End () [inline]

Referenced by [Print\(\)](#), and [Write\(\)](#).

12.272.4.8 End() [2/2]

[ConstIterator](#) gdcm::SequenceOfFragments::End () const [inline]

12.272.4.9 GetBuffer()

bool gdcm::SequenceOfFragments::GetBuffer (
 char * buffer,
 unsigned long length) const

12.272.4.10 GetFragBuffer()

bool gdcm::SequenceOfFragments::GetFragBuffer (
 unsigned int fragNb,
 char * buffer,
 unsigned long & length) const

12.272.4.11 GetFragment()

const [Fragment](#) & gdcm::SequenceOfFragments::GetFragment (
 [SizeType](#) num) const

Examples

[DecompressImage.cs](#), [FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

12.272.4.12 GetLength()

[VL](#) gdcm::SequenceOfFragments::GetLength () const [inline], [override], [virtual]

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

12.272.4.13 GetNumberOfFragments()

[SizeType](#) gdcm::SequenceOfFragments::GetNumberOfFragments () const

Examples

[FixJAIBugJPEGLS.cxx](#).

12.272.4.14 GetTable() [1/2]

[BasicOffsetTable](#) & gdcm::SequenceOfFragments::GetTable () [inline]

12.272.4.15 GetTable() [2/2]

const [BasicOffsetTable](#) & gdcm::SequenceOfFragments::GetTable () const [inline]

12.272.4.16 New()

[SmartPointer](#)< [SequenceOfFragments](#) > gdcm::SequenceOfFragments::New () [inline], [static]

Examples

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), and [MpegVideoInfo.cs](#).

References [SequenceOfFragments\(\)](#).

12.272.4.17 operator==()

bool gdcm::SequenceOfFragments::operator== (
const [Value](#) & val) const [inline], [override], [virtual]

Implements [gdcm::Value](#).

References [SequenceOfFragments\(\)](#), and [gdcm::Value::Value\(\)](#).

12.272.4.18 Print()

```
void gdcmm::SequenceOfFragments::Print (
    std::ostream & os) const    [inline], [override], [virtual]
```

Reimplemented from [gdcmm::Object](#).

References [Begin\(\)](#), [End\(\)](#), and [gdcmm_assert](#).

12.272.4.19 Read()

```
template<typename TSwap>
std::istream & gdcmm::SequenceOfFragments::Read (
    std::istream & is,
    bool readvalues = true)    [inline]
```

References [gdcmm_assert](#), [ReadPreValue\(\)](#), and [ReadValue\(\)](#).

12.272.4.20 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcmm::SequenceOfFragments::ReadPreValue (
    std::istream & is)    [inline]
```

References [gdcmmDebugMacro](#).

Referenced by [Read\(\)](#).

12.272.4.21 ReadValue()

```
template<typename TSwap>
std::istream & gdcmm::SequenceOfFragments::ReadValue (
    std::istream & is,
    bool )    [inline]
```

References [gdcmm_assert](#), [gdcmmAssertAlwaysMacro](#), [gdcmmDebugMacro](#), [gdcmmWarningMacro](#), [gdcmm::Tag::GetElement\(\)](#), [gdcmm::Tag::GetGroup\(\)](#), [gdcmm::ByteValue::GetLength\(\)](#), [gdcmm::ByteValue::GetPointer\(\)](#), [gdcmm::DataElement::GetTag\(\)](#), [gdcmm::DataElement::GetVL\(\)](#), [gdcmm::Fragment::Read\(\)](#), [gdcmm::Fragment::ReadBacktrack\(\)](#), and [gdcmm::Exception::what\(\)](#).

Referenced by [Read\(\)](#).

12.272.4.22 SetLength()

```
void gdcmm::SequenceOfFragments::SetLength (
    VL length)    [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcmm::Value](#).

12.272.4.23 Write()

```
template<typename TSwap>
std::ostream const & gdcm::SequenceOfFragments::Write (
    std::ostream & os) const    [inline]
```

References [Begin\(\)](#), [End\(\)](#), [gdcm_assert](#), [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

12.272.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
    std::ostream & os) const
```

Examples

[GetJPEGSamplePrecision.cxx](#).

The documentation for this class was generated from the following file:

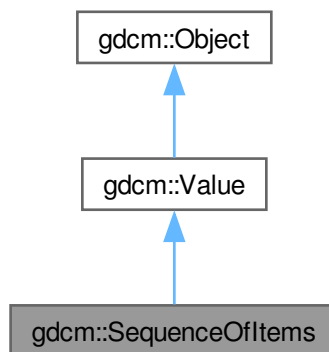
- [gdcmSequenceOfFragments.h](#)

12.273 gdcm::SequenceOfItems Class Reference

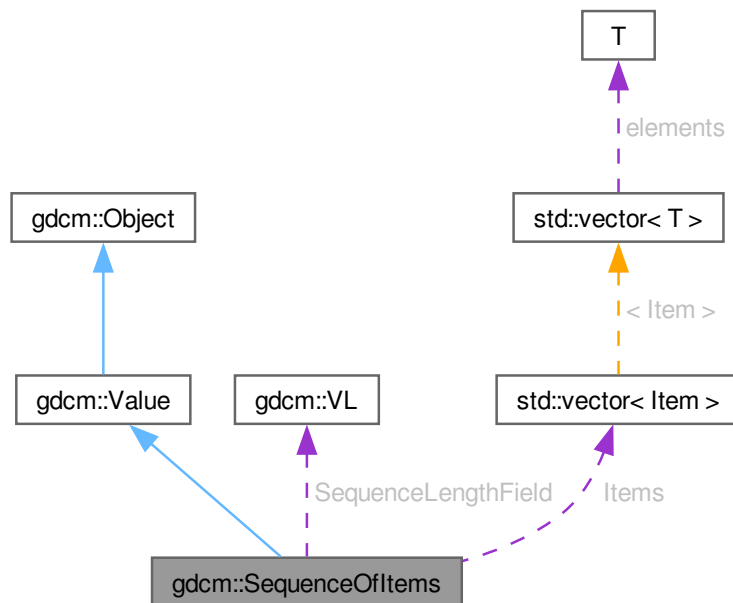
Class to represent a Sequence Of Items.

```
#include <gdcmSequenceOfItems.h>
```

Inheritance diagram for gdcm::SequenceOfItems:



Collaboration diagram for `gdc::SequenceOfItems`:



Public Types

- `typedef ItemVector::const_iterator` [ConstIterator](#)
- `typedef std::vector< Item >` [ItemVector](#)
- `typedef ItemVector::iterator` [Iterator](#)
- `typedef ItemVector::size_type` [SizeType](#)

Public Member Functions

- [SequenceOfItems](#) ()
constructor (UndefinedLength by default)
- `void` [AddItem](#) ([Item](#) const &item)
Appends an [Item](#) to the already added ones.
- [Item](#) & [AddNewUndefinedLengthItem](#) ()
Appends an [Item](#) to the already added ones.
- [Iterator](#) [Begin](#) ()
- [ConstIterator](#) [Begin](#) () const
- `void` [Clear](#) () override
remove all items within the sequence
- `template<typename TDE>`
[VL](#) [ComputeLength](#) () const

- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [Tag](#) &t) const
- [Item](#) & [GetItem](#) ([SizeType](#) position)
- const [Item](#) & [GetItem](#) ([SizeType](#) position) const
- [VL GetLength](#) () const override

Returns the SQ length, as read from disk.
- [SizeType GetNumberOfItems](#) () const
- bool [IsEmpty](#) () const
- bool [IsUndefinedLength](#) () const

return if [Value](#) Length if of undefined length
- [SequenceOfItems](#) & [operator=](#) (const [SequenceOfItems](#) &val)
- bool [operator==](#) (const [Value](#) &val) const override
- void [Print](#) (std::ostream &os) const override
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- bool [RemoveItemByIndex](#) (const [SizeType](#) index)
- void [SetLength](#) ([VL](#) length) override

Sets the actual SQ length.
- void [SetLengthToUndefined](#) ()

Properly set the Sequence of [Item](#) to be undefined length.
- void [SetNumberOfItems](#) ([SizeType](#) n)
- template<typename TDE, typename TSwap>
std::ostream const & [Write](#) (std::ostream &os) const

Public Member Functions inherited from [gdcmm::Value](#)

- [Value](#) ()=default
- [~Value](#) () override=default

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer](#)< [SequenceOfItems](#) > [New](#) ()

Public Attributes

- [ItemVector Items](#)

Vector of Sequence Items.
- [VL SequenceLengthField](#)

Total length of the Sequence (or 0xffffffff) if undefined.

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Value](#)

- virtual void [SetLengthOnly](#) (VL l)

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.273.1 Detailed Description

Class to represent a Sequence Of Items.

(value representation : SQ)

- a [Value](#) Representation for Data Elements that contains a sequence of Data Sets.
- Sequence of [Item](#) allows for Nested Data Sets

See PS 3.5, 7.4.6 Data [Element Type](#) Within a Sequence

Note

SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ) A [Value](#) Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

12.273.2 Member Typedef Documentation

12.273.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

12.273.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

12.273.2.3 Iterator

typedef ItemVector::iterator [gdcmm::SequenceOfItems::Iterator](#)

12.273.2.4 SizeType

typedef ItemVector::size_type [gdcmm::SequenceOfItems::SizeType](#)

Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [GetSubSequenceData.cxx](#).

12.273.3 Constructor & Destructor Documentation

12.273.3.1 SequenceOfItems()

[gdcmm::SequenceOfItems::SequenceOfItems](#) () [inline]

constructor (UndefinedLength by default)

References [SequenceLengthField](#).

Referenced by [New\(\)](#), [operator=\(\)](#), and [operator==\(\)](#).

12.273.4 Member Function Documentation

12.273.4.1 AddItem()

void [gdcmm::SequenceOfItems::AddItem](#) (
 [Item](#) const & item)

Appends an [Item](#) to the already added ones.

Examples

[Extracting_All_Resolution.cxx](#).

12.273.4.2 AddNewUndefinedLengthItem()

[Item](#) & [gdcmm::SequenceOfItems::AddNewUndefinedLengthItem](#) ()

Appends an [Item](#) to the already added ones.

12.273.4.3 Begin() [1/2]

[Iterator](#) `gdcmm::SequenceOfItems::Begin ()` [inline]

References [Items](#).

12.273.4.4 Begin() [2/2]

[ConstIterator](#) `gdcmm::SequenceOfItems::Begin () const` [inline]

References [Items](#).

12.273.4.5 Clear()

`void gdcmm::SequenceOfItems::Clear ()` [override], [virtual]

remove all items within the sequence

Implements [gdcmm::Value](#).

12.273.4.6 ComputeLength()

`template<typename TDE>`

[VL](#) `gdcmm::SequenceOfItems::ComputeLength () const`

12.273.4.7 End() [1/2]

[Iterator](#) `gdcmm::SequenceOfItems::End ()` [inline]

References [Items](#).

12.273.4.8 End() [2/2]

[ConstIterator](#) `gdcmm::SequenceOfItems::End () const` [inline]

References [Items](#).

12.273.4.9 FindDataElement()

`bool gdcmm::SequenceOfItems::FindDataElement (
 const Tag & t) const`

12.273.4.10 GetItem() [1/2]

[Item](#) & gdcmm::SequenceOfItems::GetItem (
 [SizeType](#) position)

12.273.4.11 GetItem() [2/2]

const [Item](#) & gdcmm::SequenceOfItems::GetItem (
 [SizeType](#) position) const

Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

12.273.4.12 GetLength()

[VL](#) gdcmm::SequenceOfItems::GetLength () const [inline], [override], [virtual]

Returns the SQ length, as read from disk.

Implements [gdcmm::Value](#).

References [SequenceLengthField](#).

Referenced by [Read\(\)](#).

12.273.4.13 GetNumberOfItems()

[SizeType](#) gdcmm::SequenceOfItems::GetNumberOfItems () const [inline]

Examples

[DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), and [GetSequenceUltrasound.cxx](#).

References [Items](#).

12.273.4.14 IsEmpty()

bool gdcmm::SequenceOfItems::IsEmpty () const [inline]

References [Items](#).

12.273.4.15 IsUndefinedLength()

bool gdcmm::SequenceOfItems::IsUndefinedLength () const [inline]

return if [Value](#) Length if of undefined length

References [SequenceLengthField](#).

12.273.4.16 New()

[SmartPointer](#)< [SequenceOfItems](#) > gdcmm::SequenceOfItems::New () [inline], [static]

Examples

[NewSequence.cs](#).

References [SequenceOfItems\(\)](#).

12.273.4.17 operator=()

[SequenceOfItems](#) & gdcmm::SequenceOfItems::operator= (
const [SequenceOfItems](#) & val) [inline]

References [SequenceOfItems\(\)](#), [Items](#), and [SequenceLengthField](#).

12.273.4.18 operator==()

bool gdcmm::SequenceOfItems::operator== (
const [Value](#) & val) const [inline], [override], [virtual]

Implements [gdcmm::Value](#).

References [SequenceOfItems\(\)](#), [gdcmm::Value::Value\(\)](#), [Items](#), and [SequenceLengthField](#).

12.273.4.19 Print()

void gdcmm::SequenceOfItems::Print (
std::ostream & os) const [inline], [override], [virtual]

Reimplemented from [gdcmm::Object](#).

References [Items](#), and [SequenceLengthField](#).

12.273.4.20 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::SequenceOfItems::Read (
    std::istream & is,
    bool readvalues = true) [inline]
```

References [gdcm::Item::Clear\(\)](#), [gdcm_assert](#), [gdcmDebugMacro](#), [gdcmWarningMacro](#), [gdcm::Exception::GetDescription\(\)](#), [GetLength\(\)](#), [gdcm::Item::GetNestedDataSet\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVL\(\)](#), [Items](#), [gdcm::Item::Read\(\)](#), [SequenceLengthField](#), and [gdcm::DataSet::Size\(\)](#).

12.273.4.21 RemoveItemByIndex()

```
bool gdcm::SequenceOfItems::RemoveItemByIndex (
    const SizeType index)
```

Remove an [Item](#) as specified by its index, if index > size, false is returned Index starts at 1 not 0

12.273.4.22 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
    VL length) [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements [gdcm::Value](#).

References [SequenceLengthField](#).

12.273.4.23 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ()
```

Properly set the Sequence of [Item](#) to be undefined length.

12.273.4.24 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
    SizeType n) [inline]
```

References [Items](#).

12.273.4.25 Write()

```
template<typename TDE, typename TSwap>
std::ostream const & gdcm::SequenceOfItems::Write (
    std::ostream & os) const [inline]
```

References [Items](#), [SequenceLengthField](#), [gdcm::Tag::Write\(\)](#), and [gdcm::VL::Write\(\)](#).

12.273.5 Member Data Documentation

12.273.5.1 Items

[ItemVector](#) `gdcmm::SequenceOfItems::Items`

Vector of Sequence Items.

Referenced by [Begin\(\)](#), [Begin\(\)](#), [End\(\)](#), [End\(\)](#), [GetNumberOfItems\(\)](#), [IsEmpty\(\)](#), [operator=\(\)](#), [operator==\(\)](#), [Print\(\)](#), [Read\(\)](#), [SetNumberOfItems\(\)](#), and [Write\(\)](#).

12.273.5.2 SequenceLengthField

[VL](#) `gdcmm::SequenceOfItems::SequenceLengthField`

Total length of the Sequence (or 0xffffffff) if undefined.

Referenced by [SequenceOfItems\(\)](#), [GetLength\(\)](#), [IsUndefinedLength\(\)](#), [operator=\(\)](#), [operator==\(\)](#), [Print\(\)](#), [Read\(\)](#), [SetLength\(\)](#), and [Write\(\)](#).

The documentation for this class was generated from the following file:

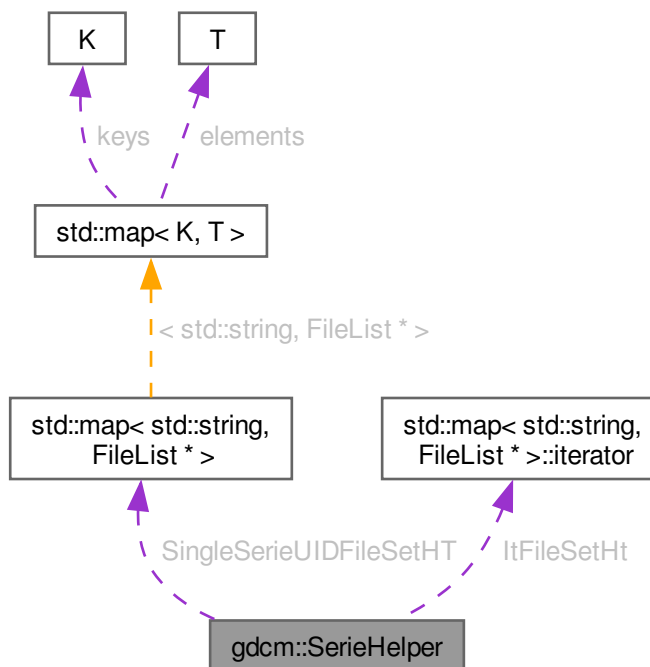
- [gdcmmSequenceOfItems.h](#)

12.274 gdcmm::SerieHelper Class Reference

[SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

```
#include <gdcmmSerieHelper.h>
```

Collaboration diagram for gdcmm::SerieHelper:



Public Member Functions

- [SerieHelper](#) ()
- [~SerieHelper](#) ()
- void [AddRestriction](#) (const std::string &tag)
- void [AddRestriction](#) (uint16_t group, uint16_t elem, std::string const &value, int op)
- void [Clear](#) ()
- void [CreateDefaultUniqueSeriesIdentifier](#) ()
- std::string [CreateUniqueSeriesIdentifier](#) (File *inFile)
- [FileList](#) * [GetFirstSingleSerieUIDFileSet](#) ()
- [FileList](#) * [GetNextSingleSerieUIDFileSet](#) ()
- void [OrderFileList](#) ([FileList](#) *fileSet)
- void [SetDirectory](#) (std::string const &dir, bool recursive=false)
- void [SetLoadMode](#) (int)
- void [SetUseSeriesDetails](#) (bool useSeriesDetails)

Protected Types

- using [Rule](#)
- typedef std::vector< [Rule](#) > [SerieRestrictions](#)
- typedef std::map< std::string, [FileList](#) * > [SingleSerieUIDFileSetmap](#)

Protected Member Functions

- bool [AddFile](#) ([FileWithName](#) &header)
- void [AddFileName](#) (std::string const &filename)
- void [AddRestriction](#) (const [Tag](#) &tag)
- bool [FileNameOrdering](#) ([FileList](#) *fileList)
- bool [ImageNumberOrdering](#) ([FileList](#) *fileList)
- bool [ImagePositionPatientOrdering](#) ([FileList](#) *fileSet)
- bool [UserOrdering](#) ([FileList](#) *fileSet)

Protected Attributes

- SingleSerieUIDFileSetmap::iterator [ItFileSetHt](#)
- [SingleSerieUIDFileSetmap](#) [SingleSerieUIDFileSetHT](#)

12.274.1 Detailed Description

[SerieHelper](#) DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Instead see [ImageHelper](#) or [IPPSorter](#)

12.274.2 Member Typedef Documentation

12.274.2.1 Rule

using [gdcmm::SerieHelper::Rule](#) [protected]

Initial value:

```
struct RuleStructure{
    uint16_t group;
    uint16_t elem;
    std::string value;
    int op;
}
```

12.274.2.2 SerieRestrictions

typedef std::vector<[Rule](#)> [gdcmm::SerieHelper::SerieRestrictions](#) [protected]

12.274.2.3 SingleSerieUIDFileSetmap

typedef std::map<std::string, [FileList](#) *> [gdcmm::SerieHelper::SingleSerieUIDFileSetmap](#) [protected]

12.274.3 Constructor & Destructor Documentation

12.274.3.1 SerieHelper()

gdcm::SerieHelper::SerieHelper ()

12.274.3.2 ~SerieHelper()

gdcm::SerieHelper::~~SerieHelper ()

12.274.4 Member Function Documentation

12.274.4.1 AddFile()

bool gdcm::SerieHelper::AddFile (
 [FileWithName](#) & header) [protected]

12.274.4.2 AddFileName()

void gdcm::SerieHelper::AddFileName (
 std::string const & filename) [protected]

12.274.4.3 AddRestriction() [1/3]

void gdcm::SerieHelper::AddRestriction (
 const std::string & tag)

12.274.4.4 AddRestriction() [2/3]

void gdcm::SerieHelper::AddRestriction (
 const [Tag](#) & tag) [protected]

12.274.4.5 AddRestriction() [3/3]

void gdcm::SerieHelper::AddRestriction (
 uint16_t group,
 uint16_t elem,
 std::string const & value,
 int op)

12.274.4.6 Clear()

```
void gdcmm::SerieHelper::Clear ()
```

12.274.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcmm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ()
```

12.274.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcmm::SerieHelper::CreateUniqueSeriesIdentifier (  
    File * inFile)
```

12.274.4.9 FileNameOrdering()

```
bool gdcmm::SerieHelper::FileNameOrdering (  
    FileList * fileList) [protected]
```

12.274.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList * gdcmm::SerieHelper::GetFirstSingleSerieUIDFileSet ()
```

12.274.4.11 GetNextSingleSerieUIDFileSet()

```
FileList * gdcmm::SerieHelper::GetNextSingleSerieUIDFileSet ()
```

12.274.4.12 ImageNumberOrdering()

```
bool gdcmm::SerieHelper::ImageNumberOrdering (  
    FileList * fileList) [protected]
```

12.274.4.13 ImagePositionPatientOrdering()

```
bool gdcmm::SerieHelper::ImagePositionPatientOrdering (  
    FileList * fileSet) [protected]
```

12.274.4.14 OrderFileList()

```
void gdcmm::SerieHelper::OrderFileList (  
    FileList * fileSet)
```

12.274.4.15 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
    std::string const & dir,
    bool recursive = false)
```

12.274.4.16 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
    int ) [inline]
```

12.274.4.17 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
    bool useSeriesDetails)
```

12.274.4.18 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
    FileList * fileSet) [protected]
```

12.274.5 Member Data Documentation

12.274.5.1 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]
```

12.274.5.2 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]
```

The documentation for this class was generated from the following file:

- [gdcmSerieHelper.h](#)

12.275 gdcm::Series Class Reference

[Series](#).

```
#include <gdcmSeries.h>
```

Public Member Functions

- [Series](#) ()=default

12.275.1 Detailed Description

[Series](#).

12.275.2 Constructor & Destructor Documentation

12.275.2.1 Series()

gdcm::Series::Series () [default]

The documentation for this class was generated from the following file:

- [gdcmSeries.h](#)

12.276 gdcm::network::ServiceClassApplicationInformation Class Reference

```
#include <gdcmServiceClassApplicationInformation.h>
```

Public Member Functions

- [ServiceClassApplicationInformation](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (uint8_t levelofsupport, uint8_t levelofdigitalsig, uint8_t elementcoercion)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.276.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

12.276.2 Constructor & Destructor Documentation

12.276.2.1 ServiceClassApplicationInformation()

gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ()

12.276.3 Member Function Documentation

12.276.3.1 Print()

```
void gdcm::network::ServiceClassApplicationInformation::Print (  
    std::ostream & os) const
```

12.276.3.2 Read()

```
std::istream & gdcm::network::ServiceClassApplicationInformation::Read (  
    std::istream & is)
```

12.276.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (  
    uint8_t levelofsupport,  
    uint8_t levelofdigitalsig,  
    uint8_t elementcoercion)
```

12.276.3.4 Size()

```
size_t gdcm::network::ServiceClassApplicationInformation::Size () const
```

12.276.3.5 Write()

```
const std::ostream & gdcm::network::ServiceClassApplicationInformation::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

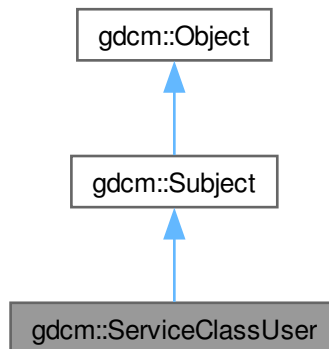
- [gdcmServiceClassApplicationInformation.h](#)

12.277 gdcmm::ServiceClassUser Class Reference

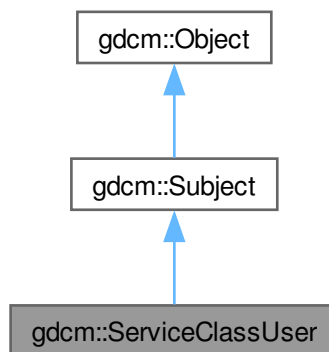
[ServiceClassUser](#).

```
#include <gdcmmServiceClassUser.h>
```

Inheritance diagram for gdcmm::ServiceClassUser:



Collaboration diagram for gdcmm::ServiceClassUser:



Public Member Functions

- [ServiceClassUser](#) ()
- [ServiceClassUser](#) (const [ServiceClassUser](#) &)=delete
- [~ServiceClassUser](#) () override
- const char * [GetAETitle](#) () const
- const char * [GetCalledAETitle](#) () const
- double [GetTimeout](#) () const
- bool [InitializeConnection](#) ()
- bool [IsPresentationContextAccepted](#) (const [PresentationContext](#) &pc) const
Return if the passed in presentation was accepted during association negotiation.
- void [operator=](#) (const [ServiceClassUser](#) &)=delete
- bool [SendEcho](#) ()
C-ECHO.
- bool [SendFind](#) (const [BaseRootQuery](#) *query, std::vector< [DataSet](#) > &retDatasets)
C-FIND a query, return result are in retDatasets.
- bool [SendMove](#) (const [BaseRootQuery](#) *query, const char *outputdir)
Execute a C-MOVE, based on query, return files are written in outputdir.
- bool [SendMove](#) (const [BaseRootQuery](#) *query, std::vector< [DataSet](#) > &retDatasets)
Execute a C-MOVE, based on query, returned dataset are Implicit.
- bool [SendMove](#) (const [BaseRootQuery](#) *query, std::vector< [File](#) > &retFile)
Execute a C-MOVE, based on query, returned Files are stored in vector.
- bool [SendStore](#) (const char *filename)
Execute a C-STORE on file on disk, named filename.
- bool [SendStore](#) ([DataSet](#) const &ds)
Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.
- bool [SendStore](#) ([File](#) const &file)
- void [SetAETitle](#) (const char *aetitle)
set calling ae title
- void [SetCalledAETitle](#) (const char *aetitle)
set called ae title
- void [SetHostname](#) (const char *hostname)
Set the name of the called hostname (hostname or IP address).
- void [SetPort](#) (uint16_t port)
Set port of remote host (called application).
- void [SetPortSCP](#) (uint16_t portscp)
Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE).
- void [SetPresentationContexts](#) (std::vector< [PresentationContext](#) > const &pcs)
Set the Presentation Context used for the Association.
- void [SetTimeout](#) (double t)
set/get Timeout
- bool [StartAssociation](#) ()
Start the association. Need to call SetPresentationContexts before.
- bool [StopAssociation](#) ()
Stop the running association.

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [ServiceClassUser](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.277.1 Detailed Description

[ServiceClassUser](#).

Examples

[CStoreQtProgress.cxx](#).

12.277.2 Constructor & Destructor Documentation

12.277.2.1 ServiceClassUser() [1/2]

gdcmm::ServiceClassUser::ServiceClassUser ()

Construct a SCU with default:

- hostname = localhost
- port = 104

Referenced by [ServiceClassUser\(\)](#), [New\(\)](#), and [operator=\(\)](#).

12.277.2.2 ~ServiceClassUser()

gdcmm::ServiceClassUser::~~ServiceClassUser () [override]

12.277.2.3 ServiceClassUser() [2/2]

gdcmm::ServiceClassUser::ServiceClassUser (
 const ServiceClassUser &) [delete]

References [ServiceClassUser\(\)](#).

12.277.3 Member Function Documentation

12.277.3.1 GetAETitle()

const char * gdcmm::ServiceClassUser::GetAETitle () const

12.277.3.2 GetCalledAETitle()

const char * gdcmm::ServiceClassUser::GetCalledAETitle () const

12.277.3.3 GetTimeout()

double gdcmm::ServiceClassUser::GetTimeout () const

12.277.3.4 InitializeConnection()

```
bool gdcm::ServiceClassUser::InitializeConnection ()
```

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

Examples

[CStoreQtProgress.cxx](#).

12.277.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
    const PresentationContext & pc) const
```

Return if the passed in presentation was accepted during association negotiation.

12.277.3.6 New()

```
SmartPointer< ServiceClassUser > gdcm::ServiceClassUser::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References [ServiceClassUser\(\)](#).

12.277.3.7 operator=()

```
void gdcm::ServiceClassUser::operator= (
    const ServiceClassUser & ) [delete]
```

References [ServiceClassUser\(\)](#).

12.277.3.8 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ()
```

C-ECHO.

12.277.3.9 SendFind()

```
bool gdcm::ServiceClassUser::SendFind (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets)
```

C-FIND a query, return result are in retDatasets.

12.277.3.10 SendMove() [1/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    const char * outputdir)
```

Execute a C-MOVE, based on query, return files are written in outputdir.

12.277.3.11 SendMove() [2/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets)
```

Execute a C-MOVE, based on query, returned dataset are Implicit.

12.277.3.12 SendMove() [3/3]

```
bool gdcm::ServiceClassUser::SendMove (
    const BaseRootQuery * query,
    std::vector< File > & retFile)
```

Execute a C-MOVE, based on query, returned Files are stored in vector.

12.277.3.13 SendStore() [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
    const char * filename)
```

Execute a C-STORE on file on disk, named filename.

Examples

[CStoreQtProgress.cxx](#).

12.277.3.14 SendStore() [2/3]

```
bool gdcm::ServiceClassUser::SendStore (
    DataSet const & ds)
```

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

12.277.3.15 SendStore() [3/3]

```
bool gdcm::ServiceClassUser::SendStore (  
    File const & file)
```

Execute a C-STORE on a [File](#), the transfer syntax used for the query is based on the file.

12.277.3.16 SetAETitle()

```
void gdcm::ServiceClassUser::SetAETitle (  
    const char * aetitle)
```

set calling ae title

12.277.3.17 SetCalledAETitle()

```
void gdcm::ServiceClassUser::SetCalledAETitle (  
    const char * aetitle)
```

set called ae title

Examples

[CStoreQtProgress.cxx](#).

12.277.3.18 SetHostname()

```
void gdcm::ServiceClassUser::SetHostname (  
    const char * hostname)
```

Set the name of the called hostname (hostname or IP address).

Examples

[CStoreQtProgress.cxx](#).

12.277.3.19 SetPort()

```
void gdcm::ServiceClassUser::SetPort (  
    uint16_t port)
```

Set port of remote host (called application).

Examples

[CStoreQtProgress.cxx](#).

12.277.3.20 SetPortSCP()

```
void gdcm::ServiceClassUser::SetPortSCP (
    uint16_t portscp)
```

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE).

12.277.3.21 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
    std::vector< PresentationContext > const & pcs)
```

Set the Presentation Context used for the Association.

Examples

[CStoreQtProgress.cxx](#).

12.277.3.22 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
    double t)
```

set/get Timeout

Examples

[CStoreQtProgress.cxx](#).

12.277.3.23 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ()
```

Start the association. Need to call SetPresentationContexts before.

Examples

[CStoreQtProgress.cxx](#).

12.277.3.24 StopAssociation()

bool gdcmm::ServiceClassUser::StopAssociation ()

Stop the running association.

Examples

[CStoreQtProgress.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmmServiceClassUser.h](#)

12.278 gdcmm::SHA1 Class Reference

Class for [SHA1](#).

```
#include <gdcmmSHA1.h>
```

Public Member Functions

- [SHA1](#) ()
- [SHA1](#) (const [SHA1](#) &)=delete
- [~SHA1](#) ()
- void [operator=](#) (const [SHA1](#) &)=delete

Static Public Member Functions

- static bool [Compute](#) (const char *buffer, unsigned long buf_len, char digest_str[20 *2+1])
- static bool [ComputeFile](#) (const char *filename, char digest_str[20 *2+1])

12.278.1 Detailed Description

Class for [SHA1](#).

Warning

this class is able to pick from one implementation:

1. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

12.278.2 Constructor & Destructor Documentation

12.278.2.1 SHA1() [1/2]

gdcm::SHA1::SHA1 ()

Referenced by [SHA1\(\)](#), and [operator=\(\)](#).

12.278.2.2 ~SHA1()

gdcm::SHA1::~~SHA1 ()

12.278.2.3 SHA1() [2/2]

gdcm::SHA1::SHA1 (
 const SHA1 &) [delete]

References [SHA1\(\)](#).

12.278.3 Member Function Documentation

12.278.3.1 Compute()

bool gdcm::SHA1::Compute (
 const char * buffer,
 unsigned long buf_len,
 char digest_str[20 *2+1]) [static]

12.278.3.2 ComputeFile()

bool gdcm::SHA1::ComputeFile (
 const char * filename,
 char digest_str[20 *2+1]) [static]

12.278.3.3 operator=()

void gdcm::SHA1::operator= (
 const [SHA1](#) &) [delete]

References [SHA1\(\)](#).

The documentation for this class was generated from the following file:

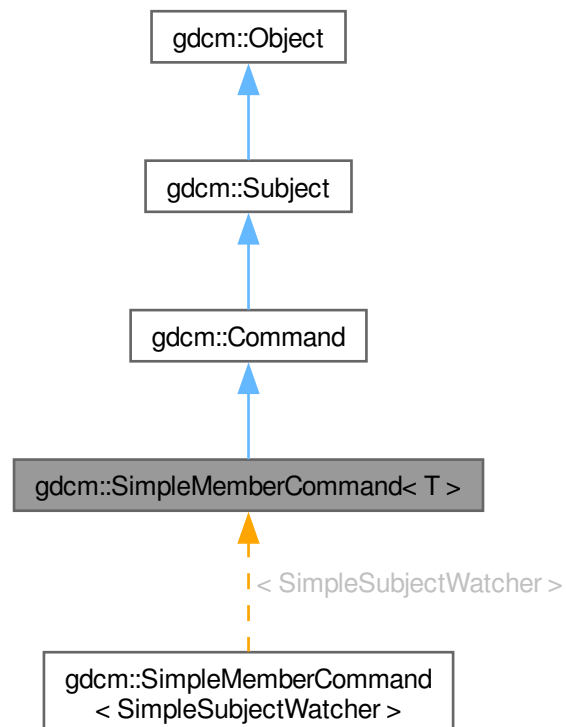
- [gdcmSHA1.h](#)

12.279 gdcM::SimpleMemberCommand< T > Class Template Reference

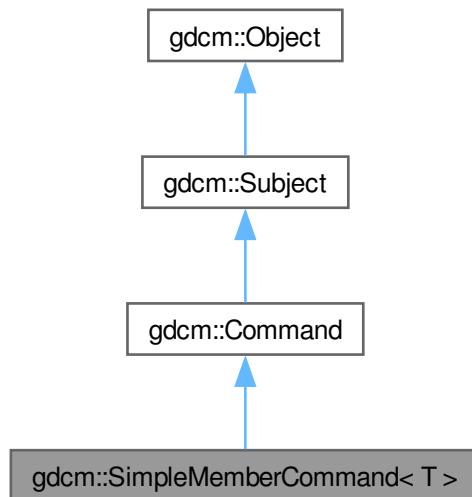
[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcMCommand.h>
```

Inheritance diagram for gdcM::SimpleMemberCommand< T >:



Collaboration diagram for gdcm::SimpleMemberCommand< T >:



Public Types

- typedef `SimpleMemberCommand Self`
- typedef `void(T::* TMemberFunctionPointer) ()`

Public Member Functions

- `SimpleMemberCommand (const Self &)=delete`
- `void Execute (const Subject *, const Event &) override`
- `void Execute (Subject *, const Event &) override`
- `void operator= (const Self &)=delete`
- `void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)`

Public Member Functions inherited from `gdcm::Command`

- `Command (const Command &)=delete`
- `void operator= (const Command &)=delete`

Public Member Functions inherited from [gdcm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [SmartPointer](#)< [SimpleMemberCommand](#) > [New](#) ()

Protected Member Functions

- [SimpleMemberCommand](#) ()
- [~SimpleMemberCommand](#) () override=default

Protected Member Functions inherited from [gdcm::Command](#)

- [Command](#) ()
- [~Command](#) () override

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Protected Attributes

- [TMemberFunctionPointer](#) [m_MemberFunction](#)
- T * [m_This](#)

12.279.1 Detailed Description

```
template<typename T>
class gdcmm::SimpleMemberCommand< T >
```

[Command](#) subclass that calls a pointer to a member function.

[SimpleMemberCommand](#) calls a pointer to a member function with no arguments.

12.279.2 Member Typedef Documentation

12.279.2.1 Self

```
template<typename T>
typedef SimpleMemberCommand gdcmm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

12.279.2.2 TMemberFunctionPointer

```
template<typename T>
typedef void(T::* gdcmm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

12.279.3 Constructor & Destructor Documentation

12.279.3.1 SimpleMemberCommand() [1/2]

```
template<typename T>
gdcmm::SimpleMemberCommand< T >::SimpleMemberCommand (
    const Self & ) [delete]
```

12.279.3.2 SimpleMemberCommand() [2/2]

```
template<typename T>
gdcmm::SimpleMemberCommand< T >::SimpleMemberCommand () [inline], [protected]
```

12.279.3.3 ~SimpleMemberCommand()

```
template<typename T>
gdcmm::SimpleMemberCommand< T >::~~SimpleMemberCommand () [override], [protected], [default]
```

12.279.4 Member Function Documentation

12.279.4.1 Execute() [1/2]

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::Execute (
    const Subject * caller,
    const Event & event) [inline], [override], [virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const [Object](#)

Implements [gdcm::Command](#).

12.279.4.2 Execute() [2/2]

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::Execute (
    Subject * ,
    const Event & ) [inline], [override], [virtual]
```

Invoke the callback function.

Implements [gdcm::Command](#).

12.279.4.3 New()

```
template<typename T>
SmartPointer< SimpleMemberCommand > gdcm::SimpleMemberCommand< T >::New () [inline], [static]
```

Run-time type information (and related methods). Method for creation through the object factory.

12.279.4.4 operator=()

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::operator= (
    const Self & ) [delete]
```

12.279.4.5 SetCallbackFunction()

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (
    T * object,
    TMemberFunctionPointer memberFunction) [inline]
```

Specify the callback function.

12.279.5 Member Data Documentation

12.279.5.1 m_MemberFunction

```
template<typename T>
T*MemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]
```

12.279.5.2 m_This

```
template<typename T>
T* gdcm::SimpleMemberCommand< T >::m_This [protected]
```

The documentation for this class was generated from the following file:

- [gdcmCommand.h](#)

12.280 gdcm::SimpleSubjectWatcher Class Reference

[SimpleSubjectWatcher](#).

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- [SimpleSubjectWatcher](#) (const SimpleSubjectWatcher &)=delete
- [SimpleSubjectWatcher](#) (Subject *s, const char *comment="")
- virtual [~SimpleSubjectWatcher](#) ()
- void [operator=](#) (const [SimpleSubjectWatcher](#) &)=delete

Protected Member Functions

- virtual void [EndFilter](#) ()
- virtual void [ShowAbort](#) ()
- virtual void [ShowAnonymization](#) (Subject *caller, const [Event](#) &evt)
- virtual void [ShowData](#) (Subject *caller, const [Event](#) &evt)
- virtual void [ShowDataSet](#) (Subject *caller, const [Event](#) &evt)
- virtual void [ShowFileName](#) (Subject *caller, const [Event](#) &evt)
- virtual void [ShowIteration](#) ()
- virtual void [ShowProgress](#) (Subject *caller, const [Event](#) &evt)
- virtual void [StartFilter](#) ()
- void [TestAbortOff](#) ()
- void [TestAbortOn](#) ()

12.280.1 Detailed Description

[SimpleSubjectWatcher](#).

This is a typical [Subject](#) Watcher class. It will observe all events.

Examples

[BasicAnonymizer.cs](#), [CStoreQtProgress.cxx](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.280.2 Constructor & Destructor Documentation

12.280.2.1 SimpleSubjectWatcher() [1/2]

```
gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    Subject * s,
    const char * comment = "")
```

Referenced by [SimpleSubjectWatcher\(\)](#), and [operator=\(\)](#).

12.280.2.2 ~SimpleSubjectWatcher()

```
virtual gdcmm::SimpleSubjectWatcher::~SimpleSubjectWatcher () [virtual]
```

12.280.2.3 SimpleSubjectWatcher() [2/2]

```
gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    const SimpleSubjectWatcher & ) [delete]
```

References [SimpleSubjectWatcher\(\)](#).

12.280.3 Member Function Documentation

12.280.3.1 EndFilter()

```
virtual void gdcmm::SimpleSubjectWatcher::EndFilter () [protected], [virtual]
```

12.280.3.2 operator=()

```
void gdcmm::SimpleSubjectWatcher::operator= (
    const SimpleSubjectWatcher & ) [delete]
```

References [SimpleSubjectWatcher\(\)](#).

12.280.3.3 ShowAbort()

virtual void gdcm::SimpleSubjectWatcher::ShowAbort () [protected], [virtual]

12.280.3.4 ShowAnonymization()

virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
 [Subject](#) * caller,
 const [Event](#) & evt) [protected], [virtual]

12.280.3.5 ShowData()

virtual void gdcm::SimpleSubjectWatcher::ShowData (
 [Subject](#) * caller,
 const [Event](#) & evt) [protected], [virtual]

12.280.3.6 ShowDataSet()

virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
 [Subject](#) * caller,
 const [Event](#) & evt) [protected], [virtual]

12.280.3.7 ShowFileName()

virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
 [Subject](#) * caller,
 const [Event](#) & evt) [protected], [virtual]

Examples

[SimpleScanner.cxx](#).

12.280.3.8 ShowIteration()

virtual void gdcm::SimpleSubjectWatcher::ShowIteration () [protected], [virtual]

12.280.3.9 ShowProgress()

virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
 [Subject](#) * caller,
 const [Event](#) & evt) [protected], [virtual]

12.280.3.10 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter () [protected], [virtual]
```

12.280.3.11 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff () [protected]
```

12.280.3.12 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn () [protected]
```

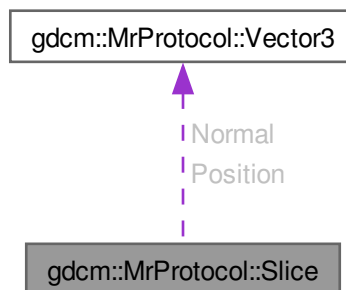
The documentation for this class was generated from the following file:

- [gdcmSimpleSubjectWatcher.h](#)

12.281 gdcm::MrProtocol::Slice Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::Slice:



Public Attributes

- [Vector3 Normal](#)
- [Vector3 Position](#)

12.281.1 Member Data Documentation

12.281.1.1 Normal

[Vector3](#) gdcm::MrProtocol::Slice::Normal

12.281.1.2 Position

[Vector3](#) gdcm::MrProtocol::Slice::Position

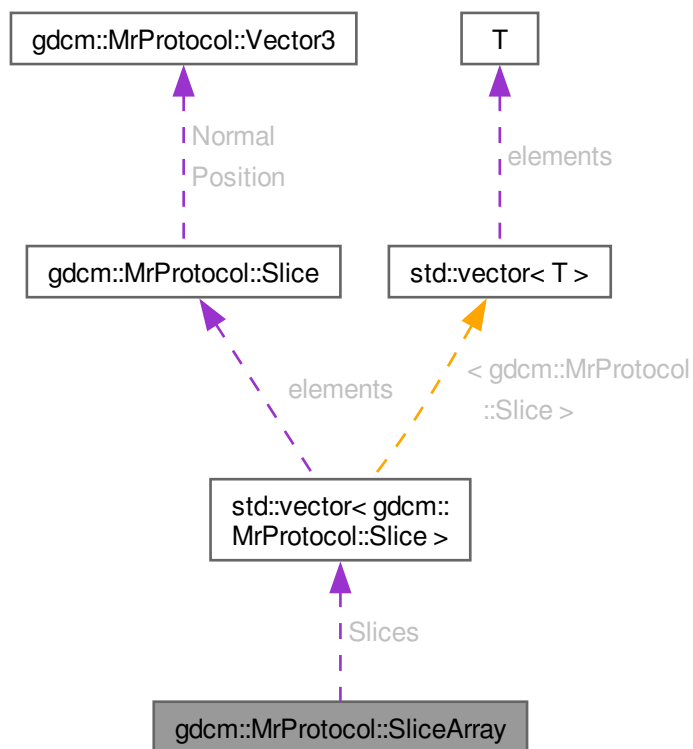
The documentation for this struct was generated from the following file:

- [gdcmMrProtocol.h](#)

12.282 gdcm::MrProtocol::SliceArray Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::SliceArray:



Public Attributes

- `std::vector< Slice > Slices`

12.282.1 Member Data Documentation

12.282.1.1 Slices

`std::vector< Slice > gdcm::MrProtocol::SliceArray::Slices`

The documentation for this struct was generated from the following file:

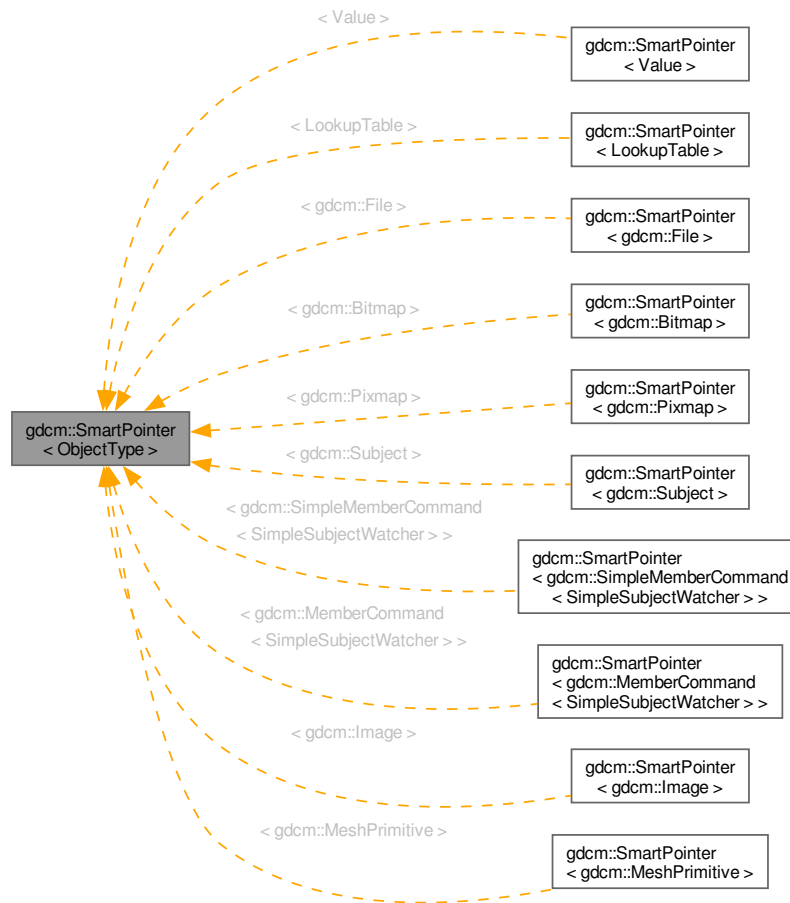
- [gdcmMrProtocol.h](#)

12.283 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

```
#include <gdcmSmartPointer.h>
```

Inheritance diagram for gdc::SmartPointer< ObjectType >:



Public Member Functions

- `SmartPointer` ()
- `SmartPointer` (const `SmartPointer< ObjectType > &p`)
- `SmartPointer` (`ObjectType *p`)
- `SmartPointer` (`ObjectType const &p`)
- `~SmartPointer` ()
- `ObjectType * GetPointer` () const
Explicit function to retrieve the pointer.
- `operator ObjectType *` () const
Return pointer to object.
- `ObjectType & operator*` () const
- `ObjectType * operator->` () const
Overload operator `->`.
- `SmartPointer & operator=` (`ObjectType *r`)

Overload operator assignment.

- [SmartPointer](#) & [operator=](#) (ObjectType const &r)
- [SmartPointer](#) & [operator=](#) ([SmartPointer](#) const &r)

Overload operator assignment.

12.283.1 Detailed Description

```
template<class ObjectType>
class gdcm::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of [gdcm::Object](#) See `tr1/shared_ptr` for a more general approach (not invasive)
`#include <tr1/memory> { shared_ptr<Bla> b(new Bla); }`

Note

Class partly based on post by Bill Hubauer: <http://groups.google.com/group/comp.lang.c++.msg/173ddc38a827a930>

See also

<http://www.davethehat.com/articles/smartp.htm>

and `itk::SmartPointer`

Examples

[CStoreQtProgress.cxx](#), [ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), [DumpVisusChange.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [SimpleScanner.cxx](#), [gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

12.283.2 Constructor & Destructor Documentation

12.283.2.1 SmartPointer() [1/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer () [inline]
```

12.283.2.2 SmartPointer() [2/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
    const SmartPointer< ObjectType > & p) [inline]
```


12.283.2.3 SmartPointer() [3/4]

```
template<class ObjectType>
gdcmm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType * p) [inline]
```

12.283.2.4 SmartPointer() [4/4]

```
template<class ObjectType>
gdcmm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType const & p) [inline]
```

12.283.2.5 ~SmartPointer()

```
template<class ObjectType>
gdcmm::SmartPointer< ObjectType >::~~SmartPointer () [inline]
```

12.283.3 Member Function Documentation

12.283.3.1 GetPointer()

```
template<class ObjectType>
ObjectType * gdcmm::SmartPointer< ObjectType >::GetPointer () const [inline]
```

Explicit function to retrieve the pointer.

12.283.3.2 operator ObjectType *()

```
template<class ObjectType>
gdcmm::SmartPointer< ObjectType >::operator ObjectType * () const [inline]
```

Return pointer to object.

12.283.3.3 operator*()

```
template<class ObjectType>
ObjectType & gdcmm::SmartPointer< ObjectType >::operator* () const [inline]
```

12.283.3.4 operator->()

```
template<class ObjectType>
ObjectType * gdcmm::SmartPointer< ObjectType >::operator-> () const [inline]
```

Overload operator ->.

12.283.3.5 `operator=()` [1/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
    ObjectType * r) [inline]
```

Overload operator assignment.

12.283.3.6 `operator=()` [2/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
    ObjectType const & r) [inline]
```

12.283.3.7 `operator=()` [3/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
    SmartPointer< ObjectType > const & r) [inline]
```

Overload operator assignment.

Referenced by [gdcm::SmartPointer< Value >::operator=\(\)](#), and [gdcm::SmartPointer< Value >::operator=\(\)](#).

The documentation for this class was generated from the following files:

- [gdcmObject.h](#)
- [gdcmSmartPointer.h](#)

12.284 `gdcm::network::SOPClassExtendedNegociationSub` Class Reference

[SOPClassExtendedNegociationSub](#).

```
#include <gdcmSOPClassExtendedNegociationSub.h>
```

Public Member Functions

- [SOPClassExtendedNegociationSub](#) ()
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetTuple](#) (const char *uid, uint8_t levelofsupport=3, uint8_t levelofdigitalsig=0, uint8_t elementcoercion=2)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.284.1 Detailed Description

[SOPClassExtendedNegociationSub](#).

PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

12.284.2 Constructor & Destructor Documentation

12.284.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ()
```

12.284.3 Member Function Documentation

12.284.3.1 Print()

```
void gdcm::network::SOPClassExtendedNegociationSub::Print (  
    std::ostream & os) const
```

12.284.3.2 Read()

```
std::istream & gdcm::network::SOPClassExtendedNegociationSub::Read (  
    std::istream & is)
```

12.284.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (  
    const char * uid,  
    uint8_t levelofsupport = 3,  
    uint8_t levelofdigitalsig = 0,  
    uint8_t elementcoercion = 2)
```

12.284.3.4 Size()

```
size_t gdcm::network::SOPClassExtendedNegociationSub::Size () const
```

12.284.3.5 Write()

```
const std::ostream & gdcm::network::SOPClassExtendedNegociationSub::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmSOPClassExtendedNegociationSub.h](#)

12.285 gdcm::SOPClassUIDToIOD Class Reference

Class convert a class SOP Class UID into [IOD](#).

```
#include <gdcmSOPClassUIDToIOD.h>
```

Public Types

- typedef const char * [const](#)(SOPClassUIDToIODType)[2]

Static Public Member Functions

- static [const](#) char * [GetIOD](#) (UIDs [const](#) &uid)
- static [const](#) char * [GetIODFromSOPClassUID](#) ([const](#) char *sopclassuid)
- static unsigned int [GetNumberOfSOPClassToIOD](#) ()
Return the number of SOP Class UID listed internally.
- static [const](#) char * [GetSOPClassUIDFromIOD](#) ([const](#) char *iod)
- static SOPClassUIDToIODType & [GetSOPClassUIDToIOD](#) (unsigned int i)
- static SOPClassUIDToIODType * [GetSOPClassUIDToIODs](#) ()

12.285.1 Detailed Description

Class convert a class SOP Class UID into [IOD](#).

Reference PS 3.4 [Table](#) B.5-1 STANDARD SOP CLASSES

12.285.2 Member Typedef Documentation

12.285.2.1 const

```
typedef const char * gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]
```

12.285.3 Member Function Documentation

12.285.3.1 GetIOD()

```
const char * gdcm::SOPClassUIDToIOD::GetIOD (  
    UIDs const & uid) [static]
```

Return the associated [IOD](#) based on a SOP Class UID uid (there is a one-to-one mapping from SOP Class UID to matching [IOD](#))

Examples

[GenerateStandardSOPClasses.cxx](#).

12.285.3.2 GetIODFromSOPClassUID()

```
const char * gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
    const char * sopclassuid) [static]
```

12.285.3.3 GetNumberOfSOPClassToIOD()

```
unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD () [static]
```

Return the number of SOP Class UID listed internally.

12.285.3.4 GetSOPClassUIDFromIOD()

```
const char * gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
    const char * iod) [static]
```

12.285.3.5 GetSOPClassUIDToIOD()

```
SOPClassUIDToIODType & gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
    unsigned int i) [static]
```

12.285.3.6 GetSOPClassUIDToIODs()

```
SOPClassUIDToIODType * gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs () [static]
```

The documentation for this class was generated from the following file:

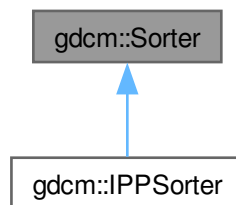
- [gdcmSOPClassUIDToIOD.h](#)

12.286 gdcm::Sorter Class Reference

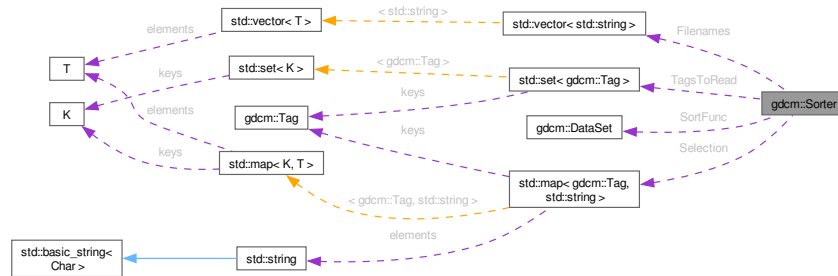
[Sorter](#).

```
#include <gdcmSorter.h>
```

Inheritance diagram for gdcm::Sorter:



Collaboration diagram for `gdc::Sorter`:



Public Types

- typedef `bool(* SortFunction) (DataSet const &, DataSet const &)`
Set the sort function which compares one dataset to the other.

Public Member Functions

- `Sorter ()`
- virtual `~Sorter ()`
- bool `AddSelect (Tag const &tag, const char *value)`
UNSUPPORTED FOR NOW.
- const `std::vector< std::string > & GetFileNames () const`
- void `Print (std::ostream &os) const`
Print.
- void `SetSortFunction (SortFunction f)`
- void `SetTagsToRead (std::set< Tag > const &tags)`
- virtual bool `Sort (std::vector< std::string > const &filenames)`
Typically the output of `Directory::GetFileNames()`.
- virtual bool `StableSort (std::vector< std::string > const &filenames)`

Protected Types

- typedef `std::map< Tag, std::string > SelectionMap`

Protected Attributes

- `std::vector< std::string > Filenames`
- `std::map< Tag, std::string > Selection`
- `SortFunction SortFunc`
- `std::set< Tag > TagsToRead`

Friends

- `std::ostream & operator<<` (`std::ostream &_os`, `const Sorter &s`)

12.286.1 Detailed Description

[Sorter](#).

General class to do sorting using a custom function You simply need to provide a function of type:↵
[Sorter::SortFunction](#)

Warning

implementation details. For now there is no cache mechanism. Which means that every time you call Sort, all files specified as input parameter are read

See also

[Scanner](#)

Examples

[SortImage.cxx](#), [SortImage2.cs](#), and [VolumeSorter.cxx](#).

12.286.2 Member Typedef Documentation

12.286.2.1 SelectionMap

```
typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap [protected]
```

12.286.2.2 SortFunction

```
typedef bool(* gdcm::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

12.286.3 Constructor & Destructor Documentation

12.286.3.1 Sorter()

```
gdcm::Sorter::Sorter ()
```

Referenced by [operator<<](#).

12.286.3.2 ~Sorter()

virtual gdcmm::Sorter::~Sorter () [virtual]

12.286.4 Member Function Documentation

12.286.4.1 AddSelect()

```
bool gdcmm::Sorter::AddSelect (  
    Tag const & tag,  
    const char * value)
```

UNSUPPORTED FOR NOW.

12.286.4.2 GetFilenames()

```
const std::vector< std::string > & gdcmm::Sorter::GetFilenames () const [inline]
```

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before [Sort\(\)](#) is called)

Examples

[Compute3DSpacing.cxx](#), [SortImage.cxx](#), [VolumeSorter.cxx](#), [gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

References [Filenames](#).

12.286.4.3 Print()

```
void gdcmm::Sorter::Print (  
    std::ostream & os) const
```

Print.

Examples

[SortImage.cxx](#), [VolumeSorter.cxx](#), and [gdcmmorthoplanes.cxx](#).

Referenced by [operator<<](#).

12.286.4.4 SetSortFunction()

```
void gdcmm::Sorter::SetSortFunction (  
    SortFunction f)
```

Examples

[SortImage.cxx](#), [SortImage2.cs](#), and [VolumeSorter.cxx](#).

12.286.4.5 SetTagsToRead()

```
void gdcmm::Sorter::SetTagsToRead (  
    std::set< Tag > const & tags)
```

Specify a set of tags to be read in during the sort procedure. By default this set is empty, in which case the entire image, including pixel data, is read in.

12.286.4.6 Sort()

```
virtual bool gdcmm::Sorter::Sort (  
    std::vector< std::string > const & filenames) [virtual]
```

Typically the output of [Directory::GetFilenames\(\)](#).

Reimplemented in [gdcmm::IPPSorter](#).

Examples

[SortImage.cxx](#).

12.286.4.7 StableSort()

```
virtual bool gdcmm::Sorter::StableSort (  
    std::vector< std::string > const & filenames) [virtual]
```

Examples

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

12.286.5 Friends And Related Symbol Documentation

12.286.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const Sorter & s) [friend]
```

References [Sorter\(\)](#), and [Print\(\)](#).

12.286.6 Member Data Documentation

12.286.6.1 Filenames

```
std::vector<std::string> gdcmm::Sorter::Filenames [protected]
```

Referenced by [GetFilenames\(\)](#).

12.286.6.2 Selection

`std::map<Tag,std::string> gdcM::Sorter::Selection` [protected]

12.286.6.3 SortFunc

[SortFunction](#) `gdcM::Sorter::SortFunc` [protected]

12.286.6.4 TagsToRead

`std::set<Tag> gdcM::Sorter::TagsToRead` [protected]

The documentation for this class was generated from the following file:

- [gdcMSorter.h](#)

12.287 gdcM::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcMSpacing.h>
```

Public Types

- enum [SpacingType](#) {
 [DETECTOR](#) = 0 ,
 [MAGNIFIED](#) ,
 [CALIBRATED](#) ,
 [UNKNOWN](#) }

Public Member Functions

- [Spacing](#) ()
- [~Spacing](#) ()=default

Static Public Member Functions

- static [Attribute](#)< 0x28, 0x34 > [ComputePixelAspectRatioFromPixelSpacing](#) (const [Attribute](#)< 0x28, 0x30 > &pixelspacing)

12.287.1 Detailed Description

Class for [Spacing](#).

It all began with a mail to WG6:

Subject: Imager Pixel [Spacing](#) vs Pixel [Spacing](#) **Body:** [Apologies for the duplicate post, namely to David Clunie & OFFIS team]

I have been trying to understand CP-586 in the following two cases:

On the one hand:

- DISCIMG/IMAGES/CRIMAGE taken from <http://dclunie.com/images/pixelspacingtestimages.zip>

And on the other hand:

- http://gdcm.sourceforge.net/thingies/cr_pixelspacing.dcm

If I understand correctly the CP, one is required to use Pixel [Spacing](#) for measurement ('true size' print) instead of Imager Pixel [Spacing](#), since the two attributes are present and Pixel [Spacing](#) is different from Imager Pixel [Spacing](#).

If this is correct, then the test data DISCIMG/IMAGES/CRIMAGE is incorrect. If this is incorrect (ie. I need to use Imager Pixel [Spacing](#)), then the display of cr_pixelspacing.dcm for measurement will be incorrect.

Could someone please let me know what am I missing here? I could not find any information in any header that would allow me to differentiate those.

Thank you for your time,

Ref: <http://lists.nema.org/scripts/lyris.pl?sub=488573&id=400720477>

See PS 3.3-2008, [Table C.7-11b](#) IMAGE PIXEL MACRO ATTRIBUTES

Ratio of the vertical size and horizontal size of the pixels in the image specified by a pair of integer values where the first value is the vertical pixel size, and the second value is the horizontal pixel size. Required if the aspect ratio values do not have a ratio of 1:1 and the physical pixel spacing is not specified by Pixel [Spacing](#) (0028,0030), or Imager Pixel [Spacing](#) (0018,1164) or Nominal Scanned Pixel [Spacing](#) (0018,2010), either for the entire [Image](#) or per-frame in a Functional Group [Macro](#). See C.7.6.3.1.7.

PS 3.3-2008 10.7.1.3 Pixel [Spacing Value](#) Order and Valid Values All pixel spacing related attributes shall have non-zero values, except when there is only a single row or column or pixel of data present, in which case the corresponding value may be zero.

Ref: http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing

12.287.2 Member Enumeration Documentation

12.287.2.1 SpacingType

enum [gdcmm::Spacing::SpacingType](#)

Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

12.287.3 Constructor & Destructor Documentation

12.287.3.1 Spacing()

[gdcmm::Spacing::Spacing](#) ()

12.287.3.2 ~Spacing()

[gdcmm::Spacing::~~Spacing](#) () [default]

12.287.4 Member Function Documentation

12.287.4.1 ComputePixelAspectRatioFromPixelSpacing()

[Attribute](#)< 0x28, 0x34 > [gdcmm::Spacing::ComputePixelAspectRatioFromPixelSpacing](#) (
const [Attribute](#)< 0x28, 0x30 > & pixelspacing) [static]

The documentation for this class was generated from the following file:

- [gdcmmSpacing.h](#)

12.288 gdcmm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmmSpectroscopy.h>
```

Public Member Functions

- [Spectroscopy](#) ()=default

12.288.1 Detailed Description

[Spectroscopy](#) class.

12.288.2 Constructor & Destructor Documentation

12.288.2.1 Spectroscopy()

gdcm::Spectroscopy::Spectroscopy () [default]

The documentation for this class was generated from the following file:

- [gdcmSpectroscopy.h](#)

12.289 gdcm::SplitMosaicFilter Class Reference

[SplitMosaicFilter](#) class.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- bool [ComputeMOSAICImagePositionPatient](#) (double pos[3], const double ipp[6], const double dircos[6], const double pixelspacing[3], const unsigned int image_dims[3], const unsigned int mosaic_dims[3], bool inverted)

Extract the value for ImagePositionPatient.

- bool [ComputeMOSAICSliceNormal](#) (double dims[3], bool &inverted)

Extract the value for SliceNormalVector (CSA header).

- bool [ComputeMOSAICSlicePosition](#) (double pos[3], bool inverted)
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [Image](#) & [GetImage](#) ()
- const [Image](#) & [GetImage](#) () const
- void [SetFile](#) (const [File](#) &f)
- void [SetImage](#) (const [Image](#) &image)
- bool [Split](#) ()

Split the SIEMENS MOSAIC image.

Static Public Member Functions

- static const [DataElement](#) & [ComputeCSAImageHeaderInfo](#) (const [DataSet](#) &ds, bool handleMissing↵ PrivateCreator=true)
Return the [DataElement](#) for the CSA [Image](#) Header.
- static const [DataElement](#) & [ComputeCSASeriesHeaderInfo](#) (const [DataSet](#) &ds, bool handleMissing↵ PrivateCreator=true)
Return the [DataElement](#) for the CSA [Series](#) Header.
- static bool [GetAcquisitionSize](#) (unsigned int size[2], [DataSet](#) const &ds)
Get the Acquisition Matrix (non zero value):
- static unsigned int [GetNumberOfImagesInMosaic](#) ([File](#) const &file)
Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

12.289.1 Detailed Description

[SplitMosaicFilter](#) class.

Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA [Image](#) Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture

Warning

when private attributes are not found, the acquisition matrix is used to compute the NumberOfImages↵ InMosaic. This means trailing black slices will be considered in the volume (instead of discarded). CSA 0029,1010 is needed for correct NumberOfImagesInMosaic CSA 0029,1020 is needed to compute the correct origin without above info default are taken (may not be accurate).

12.289.2 Constructor & Destructor Documentation

12.289.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ()
```

12.289.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~~SplitMosaicFilter ()
```

12.289.3 Member Function Documentation

12.289.3.1 ComputeCSAImageHeaderInfo()

```
const DataElement & gdcm::SplitMosaicFilter::ComputeCSAImageHeaderInfo (
    const DataSet & ds,
    bool handleMissingPrivateCreator = true) [static]
```

Return the [DataElement](#) for the CSA [Image](#) Header.

12.289.3.2 ComputeCSASeriesHeaderInfo()

```
const DataElement & gdcm::SplitMosaicFilter::ComputeCSASeriesHeaderInfo (  
    const DataSet & ds,  
    bool handleMissingPrivateCreator = true) [static]
```

Return the [DataElement](#) for the CSA [Series](#) Header.

12.289.3.3 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (  
    unsigned int dims[3])
```

Compute the new dimensions according to private information stored in the MOSAIC header.

12.289.3.4 ComputeMOSAICImagePositionPatient()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICImagePositionPatient (  
    double pos[3],  
    const double ipp[6],  
    const double dircos[6],  
    const double pixelspacing[3],  
    const unsigned int image_dims[3],  
    const unsigned int mosaic_dims[3],  
    bool inverted)
```

Extract the value for ImagePositionPatient.

12.289.3.5 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (  
    double dims[3],  
    bool & inverted)
```

Extract the value for SliceNormalVector (CSA header).

12.289.3.6 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (  
    double pos[3],  
    bool inverted)
```

Extract the value for ImagePositionPatient (requires inverted flag) Deprecated

12.289.3.7 GetAcquisitionSize()

```
bool gdcmm::SplitMosaicFilter::GetAcquisitionSize (  
    unsigned int size[2],  
    DataSet const & ds) [static]
```

Get the Acquisition Matrix (non zero value):

12.289.3.8 GetFile() [1/2]

```
File & gdcmm::SplitMosaicFilter::GetFile () [inline]
```

12.289.3.9 GetFile() [2/2]

```
const File & gdcmm::SplitMosaicFilter::GetFile () const [inline]
```

12.289.3.10 GetImage() [1/2]

```
Image & gdcmm::SplitMosaicFilter::GetImage () [inline]
```

12.289.3.11 GetImage() [2/2]

```
const Image & gdcmm::SplitMosaicFilter::GetImage () const [inline]
```

12.289.3.12 GetNumberOfImagesInMosaic()

```
unsigned int gdcmm::SplitMosaicFilter::GetNumberOfImagesInMosaic (  
    File const & file) [static]
```

Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

12.289.3.13 SetFile()

```
void gdcmm::SplitMosaicFilter::SetFile (  
    const File & f) [inline]
```

12.289.3.14 SetImage()

```
void gdcmm::SplitMosaicFilter::SetImage (  
    const Image & image)
```


12.289.3.15 Split()

bool gdcM::SplitMosaicFilter::Split ()

Split the SIEMENS MOSAIC image.

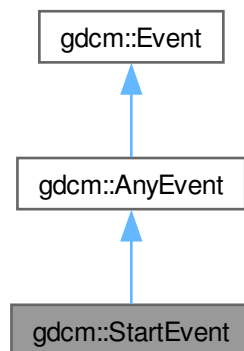
The documentation for this class was generated from the following file:

- [gdcMSplitMosaicFilter.h](#)

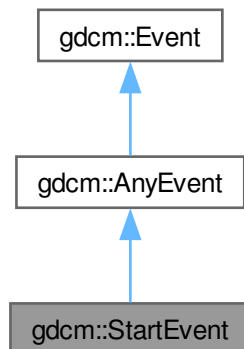
12.290 gdcM::StartEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::StartEvent:



Collaboration diagram for `gdcm::StartEvent`:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()
- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.291 `gdcm::static_assert_test< x >` Struct Template Reference

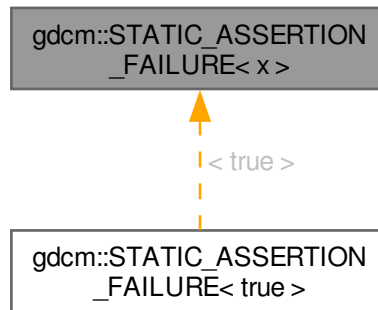
```
#include <gdcmStaticAssert.h>
```

The documentation for this struct was generated from the following file:

- [gdcmStaticAssert.h](#)

12.292 gdcmm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference

Inheritance diagram for gdcmm::STATIC_ASSERTION_FAILURE< x >:



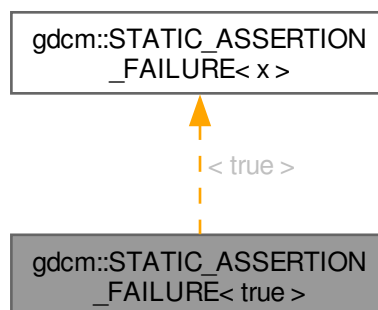
The documentation for this struct was generated from the following file:

- [gdcmmStaticAssert.h](#)

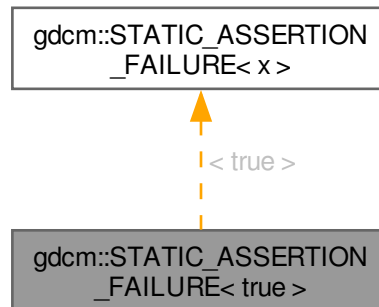
12.293 gdcmm::STATIC_ASSERTION_FAILURE< true > Struct Reference

```
#include <gdcmmStaticAssert.h>
```

Inheritance diagram for gdcmm::STATIC_ASSERTION_FAILURE< true >:



Collaboration diagram for `gdcm::STATIC_ASSERTION_FAILURE< true >`:



Public Types

- enum { `value` = 1 }

12.293.1 Member Enumeration Documentation

12.293.1.1 anonymous enum

anonymous enum

Enumerator

value	
-------	--

The documentation for this struct was generated from the following file:

- [gdcmStaticAssert.h](#)

12.294 gdcm::StreamImageReader Class Reference

[StreamImageReader](#).

```
#include <gdcmStreamImageReader.h>
```

Public Member Functions

- [StreamImageReader](#) ()
- virtual [~StreamImageReader](#) ()
- bool [CanReadImage](#) () const
- void [DefinePixelExtent](#) (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t [DefineProperBufferLength](#) () const
- std::vector< unsigned int > [GetDimensionsValueForResolution](#) (unsigned int)
- [File](#) const & [GetFile](#) () const
- bool [Read](#) (char *inReadBuffer, const std::size_t &inBufferLength)
- virtual bool [ReadImageInformation](#) ()
- void [SetFileName](#) (const char *inFileName)
- void [SetStream](#) (std::istream &inStream)

12.294.1 Detailed Description

[StreamImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

[Image](#)

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.2 Constructor & Destructor Documentation

12.294.2.1 StreamImageReader()

gdcm::StreamImageReader::StreamImageReader ()

12.294.2.2 ~StreamImageReader()

virtual gdcm::StreamImageReader::~~StreamImageReader () [virtual]

12.294.3 Member Function Documentation

12.294.3.1 CanReadImage()

```
bool gdcm::StreamImageReader::CanReadImage () const
```

Only RAW images are currently readable by the stream reader. As more streaming codecs are added, then this function will be updated to reflect those changes. Calling this function prior to reading will ensure that only streamable files are streamed. Make sure to call ReadImageInformation prior to calling this function.

Examples

[StreamImageReaderTest.cxx](#).

12.294.3.2 DefinePixelExtent()

```
void gdcm::StreamImageReader::DefinePixelExtent (
    uint16_t inXMin,
    uint16_t inXMax,
    uint16_t inYMin,
    uint16_t inYMax,
    uint16_t inZMin = 0,
    uint16_t inZMax = 1)
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^{16} pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation.

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageReader::DefineProperBufferLength () const
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char* buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.4 GetDimensionsValueForResolution()

```
std::vector< unsigned int > gdcm::StreamImageReader::GetDimensionsValueForResolution (
    unsigned int )
```

12.294.3.5 GetFile()

[File](#) const & gdcm::StreamImageReader::GetFile () const

Returns the dataset read by ReadImageInformation Couple this with the [ImageHelper](#) to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.6 Read()

```
bool gdcm::StreamImageReader::Read (
    char * inReadBuffer,
    const std::size_t & inBufferLength)
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from char* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use [ImageReader](#) instead.

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.7 ReadImageInformation()

```
virtual bool gdcm::StreamImageReader::ReadImageInformation () [virtual]
```

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel (7fe0,0010) tag.

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.8 SetFileName()

```
void gdcm::StreamImageReader::SetFileName (  
    const char * inFileName)
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal [Reader](#) class to be able to get non-pixel image information.

Examples

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

12.294.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (  
    std::istream & inStream)
```

The documentation for this class was generated from the following file:

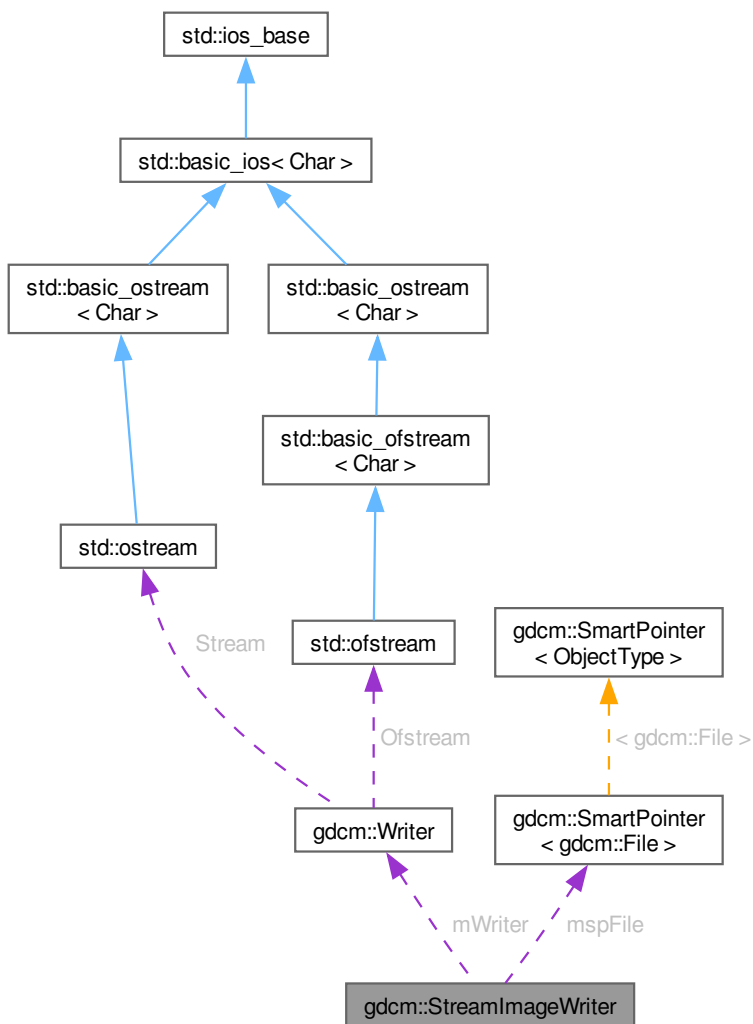
- [gdcmStreamImageReader.h](#)

12.295 gdcm::StreamImageWriter Class Reference

[StreamImageReader](#).

```
#include <gdcmStreamImageWriter.h>
```


Collaboration diagram for gdcm::StreamImageWriter:



Public Member Functions

- [StreamImageWriter](#) ()
- virtual [~StreamImageWriter](#) ()
- bool [CanWriteFile](#) () const
- void [DefinePixelExtent](#) (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t [DefineProperBufferLength](#) ()
- void [SetFile](#) (const [File](#) &inFile)
- void [SetFileName](#) (const char *inFileName)
- void [SetStream](#) (std::ostream &inStream)
- bool [Write](#) (void *inWriteBuffer, const std::size_t &inBufferLength)
- virtual bool [WriteImageInformation](#) ()

Protected Member Functions

- virtual bool [WriteImageSubregionRAW](#) (char *inWriteBuffer, const std::size_t &inBufferLength)
- int [WriteRawHeader](#) ([RAWCodec](#) *inCodec, std::ostream *inStream)

Protected Attributes

- int [mElementOffsets](#)
- int [mElementOffsets1](#)
- [SmartPointer](#)< [File](#) > [mspFile](#)
- [Writer](#) [mWriter](#)
- uint16_t [mXMax](#)
- uint16_t [mXMin](#)
- uint16_t [mYMax](#)
- uint16_t [mYMin](#)
- uint16_t [mZMax](#)
- uint16_t [mZMin](#)

12.295.1 Detailed Description

[StreamImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is threadsafe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

[Image](#)

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.2 Constructor & Destructor Documentation

12.295.2.1 [StreamImageWriter](#)()

gdcm::StreamImageWriter::StreamImageWriter ()

12.295.2.2 [~StreamImageWriter](#)()

virtual gdcm::StreamImageWriter::~~StreamImageWriter () [virtual]

12.295.3 Member Function Documentation

12.295.3.1 CanWriteFile()

```
bool gdcm::StreamImageWriter::CanWriteFile () const
```

This function determines if a file can even be written using the streaming writer unlike the reader, can be called before WriteImageInformation, but must be called after SetFile.

Examples

[Extracting_All_Resolution.cxx](#), and [Fake_Image_Using_Stream_Image_Writer.cxx](#).

12.295.3.2 DefinePixelExtent()

```
void gdcm::StreamImageWriter::DefinePixelExtent (  
    uint16_t inXMin,  
    uint16_t inXMax,  
    uint16_t inYMin,  
    uint16_t inYMax,  
    uint16_t inZMin = 0,  
    uint16_t inZMax = 1)
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^{16} pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageWriter::DefineProperBufferLength ()
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. If the return is 0, then that means that the pixel extent was not defined prior this return is for RAW inputs which are then encoded by the writer, but are used to ensure that the writer gets the proper buffer size

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.4 SetFile()

```
void gdcm::StreamImageWriter::SetFile (  
    const File & inFile)
```

Set the image information to be written to disk that is everything but the pixel information: (7fe0,0010) PixelData

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.5 SetFileName()

```
void gdcm::StreamImageWriter::SetFileName (  
    const char * inFileName)
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal [Reader](#) class to be able to get non-pixel image information.

12.295.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (  
    std::ostream & inStream)
```

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.7 Write()

```
bool gdcm::StreamImageWriter::Write (  
    void * inWriteBuffer,  
    const std::size_t & inBufferLength)
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void* to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use [ImageReader](#) instead.

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.8 WriteImageInformation()

virtual bool gdcm::StreamImageWriter::WriteImageInformation () [virtual]

Write the header information to disk, and a bunch of zeros for the actual pixel information. Of course, if we're doing a non-compressed format, that works but if it's compressed, we have to force the ordering of chunks that are written.

Examples

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.295.3.9 WriteImageSubregionRAW()

virtual bool gdcm::StreamImageWriter::WriteImageSubregionRAW (
 char * inWriteBuffer,
 const std::size_t & inBufferLength) [protected], [virtual]

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer. Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. reads by the RAW codec; other codecs are added once implemented

12.295.3.10 WriteRawHeader()

int gdcm::StreamImageWriter::WriteRawHeader (
 RAWCodec * inCodec,
 std::ostream * inStream) [protected]

when writing a raw file, we know the full extent, and can just write the first 12 bytes out (the tag, the VR, and the size) when we do compressed files, we'll do it in chunks, as described in 2009-3, part 5, Annex A, section 4. Pass the raw codec so that in the rare case of a bigendian explicit raw, the first 12 bytes written out should still be kosher. returns -1 if there's any failure, or the complete offset (12 bytes) if it works. Those 12 bytes are then added to the position in order to determine where to write.

12.295.4 Member Data Documentation

12.295.4.1 mElementOffsets

int gdcm::StreamImageWriter::mElementOffsets [protected]

The result of WriteRawHeader (or another header, when that's implemented) This result is saved so that the first N bytes aren't constantly being rewritten for each chunk that's passed in. For compressed data, the offset table will require rewrites of data.

12.295.4.2 mElementOffsets1

int gdcm::StreamImageWriter::mElementOffsets1 [protected]

12.295.4.3 mspFile

[SmartPointer<File>](#) gdcm::StreamImageWriter::mspFile [protected]

12.295.4.4 mWriter

[Writer](#) gdcm::StreamImageWriter::mWriter [protected]

12.295.4.5 mXMax

uint16_t gdcm::StreamImageWriter::mXMax [protected]

12.295.4.6 mXMin

uint16_t gdcm::StreamImageWriter::mXMin [protected]

12.295.4.7 mYMax

uint16_t gdcm::StreamImageWriter::mYMax [protected]

12.295.4.8 mYMin

uint16_t gdcm::StreamImageWriter::mYMin [protected]

12.295.4.9 mZMax

uint16_t gdcm::StreamImageWriter::mZMax [protected]

12.295.4.10 mZMin

uint16_t gdcm::StreamImageWriter::mZMin [protected]

The documentation for this class was generated from the following file:

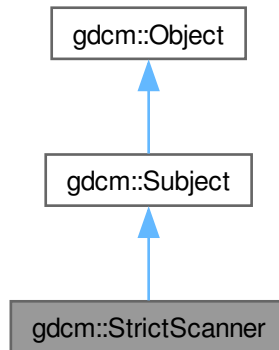
- [gdcmStreamImageWriter.h](#)

12.296 gdcm::StrictScanner Class Reference

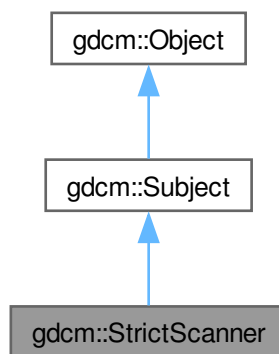
[StrictScanner](#).

```
#include <gdcmStrictScanner.h>
```

Inheritance diagram for gdcm::StrictScanner:



Collaboration diagram for gdcm::StrictScanner:



Classes

- struct [ltstr](#)

Public Types

- typedef MappingType::const_iterator [ConstIterator](#)
- typedef std::map< const char *, [TagToValue](#), ltstr > [MappingType](#)
- typedef std::map< [Tag](#), const char * > [TagToValue](#)
- typedef TagToValue::value_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [StrictScanner](#) ()
- [~StrictScanner](#) () override
- void [AddPrivateTag](#) ([PrivateTag](#) const &t)
- void [AddSkipTag](#) ([Tag](#) const &t)

Add a tag that will need to be skipped. Those are root level skip tags.
- void [AddTag](#) ([Tag](#) const &t)

Add a tag that will need to be read. Those are root level skip tags.
- [ConstIterator](#) [Begin](#) () const
- void [ClearSkipTags](#) ()
- void [ClearTags](#) ()
- [ConstIterator](#) [End](#) () const
- [Directory::FileNamesType](#) [GetAllFileNamesFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- const char * [GetFilenameFromTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- [Directory::FileNamesType](#) const & [GetFileNames](#) () const
- [Directory::FileNamesType](#) [GetKeys](#) () const
- [TagToValue](#) const & [GetMapping](#) (const char *filename) const

Get the std::map mapping filenames to value for file 'filename'.
- [TagToValue](#) const & [GetMappingFromTagToValue](#) ([Tag](#) const &t, const char *value) const

See [GetFilenameFromTagToValue\(\)](#). This is simply [GetFilenameFromTagToValue](#) followed.
- [MappingType](#) const & [GetMappings](#) () const

Mappings are the mapping from a particular tag to the map, mapping filename to value:
- [Directory::FileNamesType](#) [GetOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetValue](#) (const char *filename, [Tag](#) const &t) const
- [ValuesType](#) const & [GetValues](#) () const

Get all the values found (in lexicographic order).
- [ValuesType](#) [GetValues](#) ([Tag](#) const &t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.
- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const override

Print result.
- void [PrintTable](#) (std::ostream &os) const
- bool [Scan](#) ([Directory::FileNamesType](#) const &filenames)

Start the scan !

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer](#)< [StrictScanner](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Protected Member Functions

- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char *filename)

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [StrictScanner](#) &s)

12.296.1 Detailed Description

[StrictScanner](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a `std::set` of `std::string`. Then the address of the `cstring` underlying the `std::string` is used in the `std::map`.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.296.2 Member Typedef Documentation

12.296.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

12.296.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

12.296.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::StrictScanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

Examples

[SimpleScanner.cxx](#).

12.296.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

12.296.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

12.296.3 Constructor & Destructor Documentation

12.296.3.1 StrictScanner()

```
gdcm::StrictScanner::StrictScanner () [inline]
```

Referenced by [New\(\)](#), and [operator<<](#).

12.296.3.2 ~StrictScanner()

```
gdcm::StrictScanner::~StrictScanner () [override]
```

12.296.4 Member Function Documentation

12.296.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (  
    PrivateTag const & t)
```

12.296.4.2 AddSkipTag()

```
void gdcmm::StrictScanner::AddSkipTag (  
    Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.296.4.3 AddTag()

```
void gdcmm::StrictScanner::AddTag (  
    Tag const & t)
```

Add a tag that will need to be read. Those are root level skip tags.

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.296.4.4 Begin()

```
ConstIterator gdcmm::StrictScanner::Begin () const [inline]
```

12.296.4.5 ClearSkipTags()

```
void gdcmm::StrictScanner::ClearSkipTags ()
```

12.296.4.6 ClearTags()

```
void gdcmm::StrictScanner::ClearTags ()
```

12.296.4.7 End()

```
ConstIterator gdcmm::StrictScanner::End () const [inline]
```

12.296.4.8 GetAllFileNamesFromTagToValue()

```
Directory::FilenameType gdcmm::StrictScanner::GetAllFileNamesFromTagToValue (  
    Tag const & t,  
    const char * valuref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

12.296.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::StrictScanner::GetFilenameFromTagToValue (
    Tag const & t,
    const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.296.4.10 GetFileNames()

```
Directory::FileNamesType const & gdcm::StrictScanner::GetFileNames () const [inline]
```

12.296.4.11 GetKeys()

```
Directory::FileNamesType gdcm::StrictScanner::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.296.4.12 GetMapping()

```
TagToValue const & gdcm::StrictScanner::GetMapping (
    const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples

[SimpleScanner.cxx](#).

12.296.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::StrictScanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

12.296.4.14 GetMappings()

```
MappingType const & gdcm::StrictScanner::GetMappings () const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.296.4.15 GetOrderedValues()

[Directory::FilenameType](#) gdcmm::StrictScanner::GetOrderedValues (
[Tag](#) const & t) const

Get all the values found (in a vector) associated with [Tag](#) 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.296.4.16 GetValue()

const char * gdcmm::StrictScanner::GetValue (
 const char * filename,
[Tag](#) const & t) const

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the [TagToValue](#) hash table.

Warning

[Tag](#) 't' should have been added via [AddTag\(\)](#) prior to the [Scan\(\)](#) call !

12.296.4.17 GetValues() [1/2]

[ValuesType](#) const & gdcmm::StrictScanner::GetValues () const [inline]

Get all the values found (in lexicographic order).

12.296.4.18 GetValues() [2/2]

[ValuesType](#) gdcmm::StrictScanner::GetValues (
[Tag](#) const & t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

12.296.4.19 IsKey()

bool gdcmm::StrictScanner::IsKey (
 const char * filename) const

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.296.4.20 New()

[SmartPointer](#)< [StrictScanner](#) > gdcmm::StrictScanner::New () [inline], [static]

for wrapped language: instantiate a reference counted object

Examples

[ScanDirectory.cs](#).

References [StrictScanner\(\)](#).

12.296.4.21 Print()

```
void gdcmm::StrictScanner::Print (  
    std::ostream & os) const [override], [virtual]
```

Print result.

Reimplemented from [gdcmm::Object](#).

Referenced by [operator<<](#).

12.296.4.22 PrintTable()

```
void gdcmm::StrictScanner::PrintTable (  
    std::ostream & os) const
```

12.296.4.23 ProcessPublicTag()

```
void gdcmm::StrictScanner::ProcessPublicTag (  
    StringFilter & sf,  
    const char * filename) [protected]
```

12.296.4.24 Scan()

```
bool gdcmm::StrictScanner::Scan (  
    Directory::FileNamesType const & filenames)
```

Start the scan !

Examples

[ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.296.5 Friends And Related Symbol Documentation

12.296.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & _os,  
    const StrictScanner & s) [friend]
```

References [StrictScanner\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

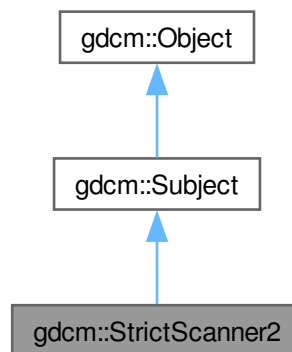
- [gdcmStrictScanner.h](#)

12.297 gdcm::StrictScanner2 Class Reference

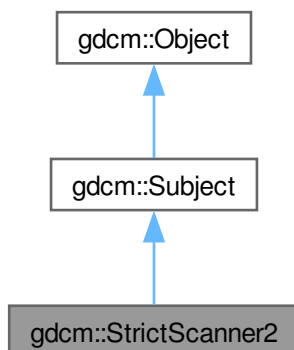
[StrictScanner2](#).

```
#include <gdcmStrictScanner2.h>
```

Inheritance diagram for gdcm::StrictScanner2:



Collaboration diagram for gdcm::StrictScanner2:



Classes

- struct [ltstr](#)

Public Types

- typedef PrivateMappingType::const_iterator [PrivateConstIterator](#)
- typedef std::map< const char *, [PrivateTagToValue](#), ltstr > [PrivateMappingType](#)
- typedef std::map< [PrivateTag](#), const char * > [PrivateTagToValue](#)
- typedef PrivateTagToValue::value_type [PrivateTagToValueValueType](#)
- typedef PublicMappingType::const_iterator [PublicConstIterator](#)
- typedef std::map< const char *, [PublicTagToValue](#), ltstr > [PublicMappingType](#)
- typedef std::map< [Tag](#), const char * > [PublicTagToValue](#)
- typedef PublicTagToValue::value_type [PublicTagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [StrictScanner2](#) ()
- [~StrictScanner2](#) () override
- bool [AddPrivateTag](#) ([PrivateTag](#) const &pt)
- bool [AddPublicTag](#) ([Tag](#) const &t)
Add a tag that will need to be read. Those are root level tags.
- bool [AddSkipTag](#) ([Tag](#) const &t)
Add a tag that will need to be skipped. Those are root level skip tags.
- [PublicConstIterator](#) [Begin](#) () const
- void [ClearPrivateTags](#) ()
- void [ClearPublicTags](#) ()

- void [ClearSkipTags](#) ()
- [PublicConstIterator End](#) () const
- [Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *valueref) const
- [Directory::FilenamesType GetAllFilenamesFromPublicTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- const char * [GetFilenameFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *valueref) const
- const char * [GetFilenameFromPublicTagToValue](#) ([Tag](#) const &t, const char *valueref) const
- [Directory::FilenamesType](#) const & [GetFilenames](#) () const

Return the list of filenames.

- [Directory::FilenamesType GetKeys](#) () const
- [PrivateTagToValue](#) const & [GetMappingFromPrivateTagToValue](#) ([PrivateTag](#) const &pt, const char *value) const
- [PublicTagToValue](#) const & [GetMappingFromPublicTagToValue](#) ([Tag](#) const &t, const char *value) const
- [PrivateTagToValue](#) const & [GetPrivateMapping](#) (const char *filename) const
- [PrivateMappingType](#) const & [GetPrivateMappings](#) () const
- [Directory::FilenamesType GetPrivateOrderedValues](#) ([PrivateTag](#) const &pt) const
- const char * [GetPrivateValue](#) (const char *filename, [PrivateTag](#) const &t) const
- [ValueType](#) [GetPrivateValues](#) ([PrivateTag](#) const &pt) const
- [PublicTagToValue](#) const & [GetPublicMapping](#) (const char *filename) const

Get the std::map mapping filenames to value for file 'filename'.

- [PublicMappingType](#) const & [GetPublicMappings](#) () const
- [Directory::FilenamesType GetPublicOrderedValues](#) ([Tag](#) const &t) const
- const char * [GetPublicValue](#) (const char *filename, [Tag](#) const &t) const
- [ValueType](#) [GetPublicValues](#) ([Tag](#) const &t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

- [ValueType](#) const & [GetValues](#) () const

Get all the values found (in lexicographic order).

- bool [IsKey](#) (const char *filename) const
- void [Print](#) (std::ostream &os) const override

Print result.

- void [PrintTable](#) (std::ostream &os, bool header=false) const

Print result as CSV table.

- [PrivateConstIterator PrivateBegin](#) () const
- [PrivateConstIterator PrivateEnd](#) () const
- bool [Scan](#) ([Directory::FilenamesType](#) const &filenames)

Start the scan !

Public Member Functions inherited from [gdcmm::Subject](#)

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)
- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Static Public Member Functions

- static [SmartPointer](#)< [StrictScanner2](#) > [New](#) ()
for wrapped language: instantiate a reference counted object

Protected Member Functions

- void [ProcessPrivateTag](#) ([StringFilter](#) &sf, const char *filename)
- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char *filename)

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [StrictScanner2](#) &s)

12.297.1 Detailed Description

[StrictScanner2](#).

This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [StringFilter](#)

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- [ProgressEvent](#)
- [StartEvent](#)
- [EndEvent](#)

12.297.2 Member Typedef Documentation

12.297.2.1 PrivateConstIterator

typedef PrivateMappingType::const_iterator [gdcm::StrictScanner2::PrivateConstIterator](#)

12.297.2.2 PrivateMappingType

typedef std::map<const char *, [PrivateTagToValue](#), [ltstr](#)> [gdcm::StrictScanner2::PrivateMappingType](#)

12.297.2.3 PrivateTagToValue

typedef std::map<[PrivateTag](#), const char *> [gdcm::StrictScanner2::PrivateTagToValue](#)

12.297.2.4 PrivateTagToValueValueType

typedef PrivateTagToValue::value_type [gdcm::StrictScanner2::PrivateTagToValueValueType](#)

12.297.2.5 PublicConstIterator

typedef PublicMappingType::const_iterator [gdcm::StrictScanner2::PublicConstIterator](#)

12.297.2.6 PublicMappingType

typedef std::map<const char *, [PublicTagToValue](#), [ltstr](#)> [gdcm::StrictScanner2::PublicMappingType](#)

12.297.2.7 PublicTagToValue

typedef std::map<[Tag](#), const char *> [gdcm::StrictScanner2::PublicTagToValue](#)

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. [Tag](#) are used as [Tag](#) class since sizeof(tag) <= sizeof(pointer)

12.297.2.8 PublicTagToValueValueType

typedef PublicTagToValue::value_type [gdcm::StrictScanner2::PublicTagToValueValueType](#)

12.297.2.9 ValuesType

typedef std::set<std::string> [gdcm::StrictScanner2::ValuesType](#)

12.297.3 Constructor & Destructor Documentation

12.297.3.1 StrictScanner2()

`gdcmm::StrictScanner2::StrictScanner2 ()` [inline]

Referenced by [New\(\)](#), and [operator<<](#).

12.297.3.2 ~StrictScanner2()

`gdcmm::StrictScanner2::~~StrictScanner2 ()` [override]

12.297.4 Member Function Documentation

12.297.4.1 AddPrivateTag()

`bool gdcmm::StrictScanner2::AddPrivateTag (`
 [PrivateTag](#) const & pt)

12.297.4.2 AddPublicTag()

`bool gdcmm::StrictScanner2::AddPublicTag (`
 [Tag](#) const & t)

Add a tag that will need to be read. Those are root level tags.

12.297.4.3 AddSkipTag()

`bool gdcmm::StrictScanner2::AddSkipTag (`
 [Tag](#) const & t)

Add a tag that will need to be skipped. Those are root level skip tags.

12.297.4.4 Begin()

[PublicConstIterator](#) `gdcmm::StrictScanner2::Begin ()` const [inline]

12.297.4.5 ClearPrivateTags()

`void gdcmm::StrictScanner2::ClearPrivateTags ()`

12.297.4.6 ClearPublicTags()

```
void gdcmm::StrictScanner2::ClearPublicTags ()
```

12.297.4.7 ClearSkipTags()

```
void gdcmm::StrictScanner2::ClearSkipTags ()
```

12.297.4.8 End()

```
PublicConstIterator gdcmm::StrictScanner2::End () const [inline]
```

12.297.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcmm::StrictScanner2::GetAllFilenamesFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valuref) const
```

12.297.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcmm::StrictScanner2::GetAllFilenamesFromPublicTagToValue (
    Tag const & t,
    const char * valuref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

12.297.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcmm::StrictScanner2::GetFilenameFromPrivateTagToValue (
    PrivateTag const & pt,
    const char * valuref) const
```

12.297.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcmm::StrictScanner2::GetFilenameFromPublicTagToValue (
    Tag const & t,
    const char * valuref) const
```

Will loop over all files and return the first file where value match the reference value 'valuref'

12.297.4.13 GetFilenames()

[Directory::FilenamesType](#) const & gdcm::StrictScanner2::GetFilenames () const [inline]

Return the list of filenames.

12.297.4.14 GetKeys()

[Directory::FilenamesType](#) gdcm::StrictScanner2::GetKeys () const

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.297.4.15 GetMappingFromPrivateTagToValue()

[PrivateTagToValue](#) const & gdcm::StrictScanner2::GetMappingFromPrivateTagToValue (
 [PrivateTag](#) const & pt,
 const char * value) const

12.297.4.16 GetMappingFromPublicTagToValue()

[PublicTagToValue](#) const & gdcm::StrictScanner2::GetMappingFromPublicTagToValue (
 [Tag](#) const & t,
 const char * value) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed

12.297.4.17 GetPrivateMapping()

[PrivateTagToValue](#) const & gdcm::StrictScanner2::GetPrivateMapping (
 const char * filename) const

12.297.4.18 GetPrivateMappings()

[PrivateMappingType](#) const & gdcm::StrictScanner2::GetPrivateMappings () const [inline]

12.297.4.19 GetPrivateOrderedValues()

[Directory::FilenamesType](#) gdcm::StrictScanner2::GetPrivateOrderedValues (
 [PrivateTag](#) const & pt) const

12.297.4.20 GetPrivateValue()

```
const char * gdcmm::StrictScanner2::GetPrivateValue (
    const char * filename,
    PrivateTag const & t) const
```

12.297.4.21 GetPrivateValues()

```
ValueType gdcmm::StrictScanner2::GetPrivateValues (
    PrivateTag const & pt) const
```

Get all the values found (in lexicographic order) associated with PrivateTag 'pt'

12.297.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcmm::StrictScanner2::GetPublicMapping (
    const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

12.297.4.23 GetPublicMappings()

```
PublicMappingType const & gdcmm::StrictScanner2::GetPublicMappings () const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.297.4.24 GetPublicOrderedValues()

```
Directory::FileNamesType gdcmm::StrictScanner2::GetPublicOrderedValues (
    Tag const & t) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.297.4.25 GetPublicValue()

```
const char * gdcmm::StrictScanner2::GetPublicValue (
    const char * filename,
    Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.297.4.26 GetPublicValues()

[ValueType](#) gdcm::StrictScanner2::GetPublicValues (
 [Tag](#) const & t) const

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

12.297.4.27 GetValues()

[ValueType](#) const & gdcm::StrictScanner2::GetValues () const [inline]

Get all the values found (in lexicographic order).

12.297.4.28 IsKey()

bool gdcm::StrictScanner2::IsKey (
 const char * filename) const

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

12.297.4.29 New()

[SmartPointer](#)< [StrictScanner2](#) > gdcm::StrictScanner2::New () [inline], [static]

for wrapped language: instantiate a reference counted object

References [StrictScanner2\(\)](#).

12.297.4.30 Print()

void gdcm::StrictScanner2::Print (
 std::ostream & os) const [override], [virtual]

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by [operator<<](#).

12.297.4.31 PrintTable()

void gdcm::StrictScanner2::PrintTable (
 std::ostream & os,
 bool header = false) const

Print result as CSV table.

12.297.4.32 PrivateBegin()

[PrivateConstIterator](#) gdcmm::StrictScanner2::PrivateBegin () const [inline]

12.297.4.33 PrivateEnd()

[PrivateConstIterator](#) gdcmm::StrictScanner2::PrivateEnd () const [inline]

12.297.4.34 ProcessPrivateTag()

```
void gdcmm::StrictScanner2::ProcessPrivateTag (  
    StringFilter & sf,  
    const char * filename) [protected]
```

12.297.4.35 ProcessPublicTag()

```
void gdcmm::StrictScanner2::ProcessPublicTag (  
    StringFilter & sf,  
    const char * filename) [protected]
```

12.297.4.36 Scan()

```
bool gdcmm::StrictScanner2::Scan (  
    Directory::FileNamesType const & filenames)
```

Start the scan !

12.297.5 Friends And Related Symbol Documentation

12.297.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const StrictScanner2 & s) [friend]
```

References [StrictScanner2\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

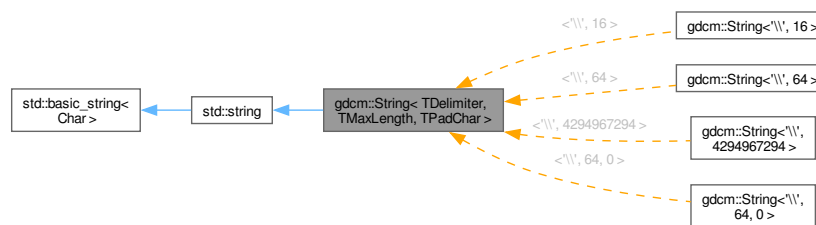
- [gdcmmStrictScanner2.h](#)

12.298 gdcmm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

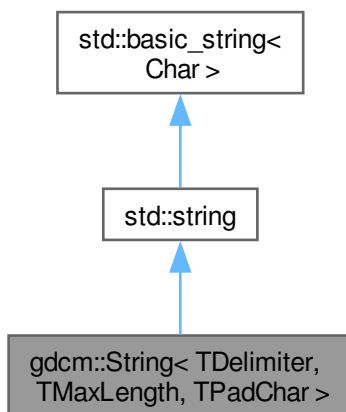
[String](#).

```
#include <gdcmmString.h>
```

Inheritance diagram for gdcmm::String< TDelimiter, TMaxLength, TPadChar >:



Collaboration diagram for gdcmm::String< TDelimiter, TMaxLength, TPadChar >:



Public Types

- typedef `std::string::const_iterator` [const_iterator](#)
- typedef `std::string::const_reference` [const_reference](#)
- typedef `std::string::const_reverse_iterator` [const_reverse_iterator](#)
- typedef `std::string::difference_type` [difference_type](#)

- typedef std::string::iterator [iterator](#)
- typedef std::string::pointer [pointer](#)
- typedef std::string::reference [reference](#)
- typedef std::string::reverse_iterator [reverse_iterator](#)
- typedef std::string::size_type [size_type](#)
- typedef std::string::value_type [value_type](#)

Public Member Functions

- [String](#) ()
[String](#) constructors.
- [String](#) (const std::string &s, [size_type](#) pos=0, [size_type](#) n=npos)
- [String](#) (const [value_type](#) *s)
- [String](#) (const [value_type](#) *s, [size_type](#) n)
- bool [IsValid](#) () const
return if string is valid
- [operator const char *](#) () const
WARNING: Trailing \0 might be lost in this operation:
- std::string [Trim](#) () const
- [gdcmm::String](#)< TDelimiter, TMaxLength, TPadChar > [Truncate](#) () const

Static Public Member Functions

- static std::string [Trim](#) (const char *input)

12.298.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcmm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. No one actually respect the max length TPadChar is the string padding (0 or space)

12.298.2 Member Typedef Documentation

12.298.2.1 const_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator
```

12.298.2.2 const_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reference
```

12.298.2.3 const_reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator
```

12.298.2.4 difference_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::difference_type
```

12.298.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

12.298.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcmm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

12.298.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

12.298.2.8 reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::reverse_iterator
```

12.298.2.9 size_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcmm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

12.298.2.10 value_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

12.298.3 Constructor & Destructor Documentation

12.298.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String () [inline]
```

[String](#) constructors.

12.298.3.2 String() [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value\_type * s) [inline]
```

12.298.3.3 String() [3/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value\_type * s,
    size\_type n) [inline]
```

12.298.3.4 String() [4/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const std::string & s,
    size\_type pos = 0,
    size\_type n = npos) [inline]
```

12.298.4 Member Function Documentation

12.298.4.1 IsValid()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid () const [inline]
```

return if string is valid

Referenced by [gdcm::String<'\\', 16 >::Truncate\(\)](#).

12.298.4.2 operator const char *()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * () const [inline]
```

WARNING: Trailing \0 might be lost in this operation:

12.298.4.3 Trim() [1/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim () const [inline]
```

Trim function is required to return a std::string object, otherwise we could not create a [gdcm::String](#) object with an odd number of bytes...

12.298.4.4 Trim() [2/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (
    const char * input) [inline], [static]
```

12.298.4.5 Truncate()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar > gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate ()
const [inline]
```

The documentation for this class was generated from the following file:

- [gdcmString.h](#)

12.299 gdcm::StringFilter Class Reference

[StringFilter](#).

```
#include <gdcmStringFilter.h>
```

Public Member Functions

- [StringFilter](#) ()
- [~StringFilter](#) ()
- bool [ExecuteQuery](#) (std::string const &query, std::string &value) const
- std::string [FromString](#) (const [Tag](#) &t, const char *value, size_t len)
Convert to string the char array defined by the pair (value,len).
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- void [SetDicts](#) (const [Dicts](#) &dicts)
Allow user to pass in there own dicts.
- void [SetFile](#) (const [File](#) &f)
Set/Get [File](#).
- std::string [ToString](#) (const [DataElement](#) &de) const
- std::string [ToString](#) (const [PrivateTag](#) &t) const
- std::string [ToString](#) (const [Tag](#) &t) const
Directly from a [Tag](#):
- std::pair< std::string, std::string > [ToStringPair](#) (const [DataElement](#) &de) const
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t) const
Directly from a [Tag](#):
- void [UseDictAlways](#) (bool)

Protected Member Functions

- bool [ExecuteQuery](#) (std::string const &query, [DataSet](#) const &ds, std::string &value) const
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t, [DataSet](#) const &ds) const

12.299.1 Detailed Description

[StringFilter](#).

[StringFilter](#) is the class that make gdc2.x looks more like gdc1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language

Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

12.299.2 Constructor & Destructor Documentation

12.299.2.1 StringFilter()

gdc2::StringFilter::StringFilter ()

12.299.2.2 ~StringFilter()

gdcm::StringFilter::~~StringFilter ()

12.299.3 Member Function Documentation

12.299.3.1 ExecuteQuery() [1/2]

```
bool gdcm::StringFilter::ExecuteQuery (
    std::string const & query,
    DataSet const & ds,
    std::string & value) const [protected]
```

12.299.3.2 ExecuteQuery() [2/2]

```
bool gdcm::StringFilter::ExecuteQuery (
    std::string const & query,
    std::string & value) const
```

Execute the XPATH query to find a value (as string) return false when attribute is not found (or an error in the XPATH query) You need to make sure that your XPATH query is syntactically correct

12.299.3.3 FromString()

```
std::string gdcm::StringFilter::FromString (
    const Tag & t,
    const char * value,
    size_t len)
```

Convert to string the char array defined by the pair (value,len).

12.299.3.4 GetFile() [1/2]

File & gdcm::StringFilter::GetFile () [inline]

12.299.3.5 GetFile() [2/2]

const File & gdcm::StringFilter::GetFile () const [inline]

12.299.3.6 SetDicts()

```
void gdcm::StringFilter::SetDicts (
    const Dicts & dicts)
```

Allow user to pass in there own dicts.

12.299.3.7 SetFile()

```
void gdcm::StringFilter::SetFile (
    const File & f) [inline]
```

Set/Get [File](#).

Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

12.299.3.8 ToString() [1/3]

```
std::string gdcm::StringFilter::ToString (
    const DataElement & de) const
```

Convert to string the [ByteValue](#) contained in a [DataElement](#). The [DataElement](#) must be coming from the actual [DataSet](#) associated with [File](#) (see SetFile).

Examples

[DumpVisusChange.cxx](#), [ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

12.299.3.9 ToString() [2/3]

```
std::string gdcm::StringFilter::ToString (
    const PrivateTag & t) const
```

12.299.3.10 ToString() [3/3]

```
std::string gdcm::StringFilter::ToString (
    const Tag & t) const
```

Directly from a [Tag](#):

12.299.3.11 ToStringPair() [1/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
    const DataElement & de) const
```

Convert to string the [ByteValue](#) contained in a [DataElement](#) the returned elements are: pair.first : the name as found in the dictionary of [DataElement](#) pari.second : the value encoded into a string (US,UL...) are properly converted

Examples

[ReadAndPrintAttributes.cxx](#).

12.299.3.12 ToStringPair() [2/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
    const Tag & t) const
```

Directly from a [Tag](#):

12.299.3.13 ToStringPair() [3/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
    const Tag & t,
    DataSet const & ds) const    [protected]
```

12.299.3.14 UseDictAlways()

```
void gdcm::StringFilter::UseDictAlways (
    bool )    [inline]
```

The documentation for this class was generated from the following file:

- [gdcmStringFilter.h](#)

12.300 gdcm::Study Class Reference

[Study](#).

```
#include <gdcmStudy.h>
```

Public Member Functions

- [Study](#) ()=default

12.300.1 Detailed Description

[Study](#).

12.300.2 Constructor & Destructor Documentation

12.300.2.1 Study()

```
gdcm::Study::Study ()    [default]
```

The documentation for this class was generated from the following file:

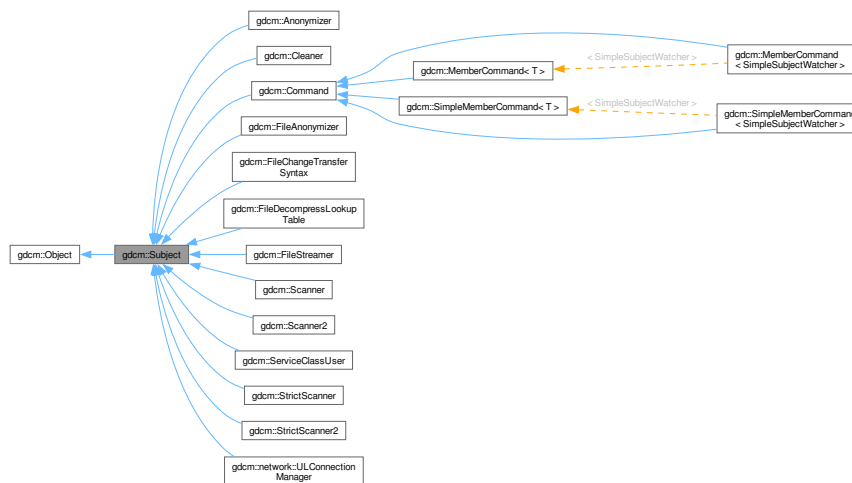
- [gdcmStudy.h](#)

12.301 gdcmm::Subject Class Reference

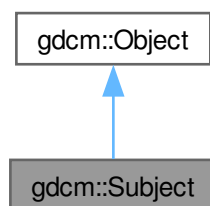
[Subject](#).

```
#include <gdcmmSubject.h>
```

Inheritance diagram for gdcmm::Subject:



Collaboration diagram for gdcmm::Subject:



Public Member Functions

- [Subject](#) ()
- [~Subject](#) () override
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *)
- unsigned long [AddObserver](#) (const [Event](#) &event, [Command](#) *) const
- [Command](#) * [GetCommand](#) (unsigned long tag)

- bool [HasObserver](#) (const [Event](#) &event) const
- void [InvokeEvent](#) (const [Event](#) &)
- void [InvokeEvent](#) (const [Event](#) &) const
- void [RemoveAllObservers](#) ()
- void [RemoveObserver](#) (unsigned long tag)

Public Member Functions inherited from [gdcmm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from [gdcmm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.301.1 Detailed Description

[Subject](#).

See also

[Command Event](#)

Examples

[BasicAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [ScanDirectory.cs](#), and [SimpleScanner.cxx](#).

12.301.2 Constructor & Destructor Documentation

12.301.2.1 [Subject](#)()

[gdcmm::Subject::Subject](#) ()

Referenced by [gdcmm::Command::Execute\(\)](#), and [gdcmm::Command::Execute\(\)](#).

12.301.2.2 ~Subject()

```
gdcmm::Subject::~~Subject () [override]
```

12.301.3 Member Function Documentation

12.301.3.1 AddObserver() [1/2]

```
unsigned long gdcmm::Subject::AddObserver (
    const Event & event,
    Command * )
```

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an [gdcmm::Command](#) to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the [Command](#) becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

12.301.3.2 AddObserver() [2/2]

```
unsigned long gdcmm::Subject::AddObserver (
    const Event & event,
    Command * ) const
```

12.301.3.3 GetCommand()

```
Command * gdcmm::Subject::GetCommand (
    unsigned long tag)
```

Get the command associated with the given tag. NOTE: This returns a pointer to a [Command](#), but it is safe to assign this to a [Command::Pointer](#). Since [Command](#) inherits from [LightObject](#), at this point in the code, only a pointer or a reference to the [Command](#) can be used.

12.301.3.4 HasObserver()

```
bool gdcmm::Subject::HasObserver (
    const Event & event) const
```

Return true if an observer is registered for this event.

12.301.3.5 InvokeEvent() [1/2]

```
void gdcmm::Subject::InvokeEvent (
    const Event & )
```

Call [Execute](#) on all the [Commands](#) observing this event id.

12.301.3.6 InvokeEvent() [2/2]

```
void gdcm::Subject::InvokeEvent (
    const Event & ) const
```

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

12.301.3.7 RemoveAllObservers()

```
void gdcm::Subject::RemoveAllObservers ()
```

Remove all observers .

12.301.3.8 RemoveObserver()

```
void gdcm::Subject::RemoveObserver (
    unsigned long tag)
```

Remove the observer with this tag value.

The documentation for this class was generated from the following file:

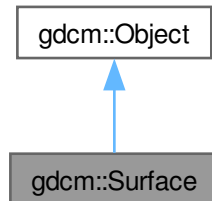
- [gdcmSubject.h](#)

12.302 gdcm::Surface Class Reference

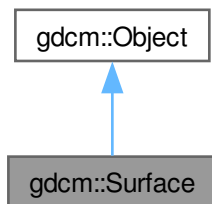
This class defines a SURFACE IE.

```
#include <gdcmSurface.h>
```

Inheritance diagram for gdcm::Surface:



Collaboration diagram for `gdcM::Surface`:



Public Types

- enum `STATES` {
`NO` = 0 ,
`YES` ,
`UNKNOWN` ,
`STATES_END` }
- enum `VIEWType` {
`SURFACE` = 0 ,
`WIREFRAME` ,
`POINTS` ,
`VIEWType_END` }

Enumeration for Recommended Presentation `Type`.

Public Member Functions

- `Surface ()`
- `~Surface ()` override
- `SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()`
- `SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily () const`
- `const char * GetAlgorithmName () const`
- `const char * GetAlgorithmVersion () const`
- `const float * GetAxisOfRotation () const`
- `const float * GetCenterOfRotation () const`
- `STATES GetFiniteVolume () const`
- `STATES GetManifold () const`
- `float GetMaximumPointDistance () const`
- `float GetMeanPointDistance () const`
- `MeshPrimitive & GetMeshPrimitive ()`
- `MeshPrimitive const & GetMeshPrimitive () const`
- `unsigned long GetNumberOfSurfacePoints () const`
- `unsigned long GetNumberOfVectors () const`
- `DataElement & GetPointCoordinatesData ()`

- const [DataElement](#) & [GetPointCoordinatesData](#) () const
- const float * [GetPointPositionAccuracy](#) () const
- const float * [GetPointsBoundingBoxCoordinates](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetProcessingAlgorithm](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetProcessingAlgorithm](#) () const
- const unsigned short * [GetRecommendedDisplayCIELabValue](#) () const
- unsigned short [GetRecommendedDisplayCIELabValue](#) (const unsigned int idx) const
- unsigned short [GetRecommendedDisplayGrayscaleValue](#) () const
- float [GetRecommendedPresentationOpacity](#) () const
- [VIEWType](#) [GetRecommendedPresentationType](#) () const
- const char * [GetSurfaceComments](#) () const
- unsigned long [GetSurfaceNumber](#) () const
- bool [GetSurfaceProcessing](#) () const
- const char * [GetSurfaceProcessingDescription](#) () const
- float [GetSurfaceProcessingRatio](#) () const
- const float * [GetVectorAccuracy](#) () const
- [DataElement](#) & [GetVectorCoordinateData](#) ()
- const [DataElement](#) & [GetVectorCoordinateData](#) () const
- unsigned short [GetVectorDimensionality](#) () const
- void [SetAlgorithmFamily](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetAlgorithmName](#) (const char *str)
- void [SetAlgorithmVersion](#) (const char *str)
- void [SetAxisOfRotation](#) (const float *axis)
- void [SetCenterOfRotation](#) (const float *center)
- void [SetFiniteVolume](#) ([STATES](#) state)
- void [SetManifold](#) ([STATES](#) state)
- void [SetMaximumPointDistance](#) (float maximum)
- void [SetMeanPointDistance](#) (float average)
- void [SetMeshPrimitive](#) ([MeshPrimitive](#) const &mp)
- void [SetNumberOfSurfacePoints](#) (const unsigned long nb)
- void [SetNumberOfVectors](#) (const unsigned long nb)
- void [SetPointCoordinatesData](#) ([DataElement](#) const &de)
- void [SetPointPositionAccuracy](#) (const float *accuracies)
- void [SetPointsBoundingBoxCoordinates](#) (const float *coordinates)
- void [SetProcessingAlgorithm](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetRecommendedDisplayCIELabValue](#) (const std::vector< unsigned short > &vl)
- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl, const unsigned int idx=0)
- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl[3])
- void [SetRecommendedDisplayGrayscaleValue](#) (const unsigned short vl)
- void [SetRecommendedPresentationOpacity](#) (const float opacity)
- void [SetRecommendedPresentationType](#) ([VIEWType](#) type)
- void [SetSurfaceComments](#) (const char *comment)
- void [SetSurfaceNumber](#) (const unsigned long nb)
- void [SetSurfaceProcessing](#) (bool b)
- void [SetSurfaceProcessingDescription](#) (const char *description)
- void [SetSurfaceProcessingRatio](#) (const float ratio)
- void [SetVectorAccuracy](#) (const float *accuracy)
- void [SetVectorCoordinateData](#) ([DataElement](#) const &de)
- void [SetVectorDimensionality](#) (const unsigned short dim)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)
- virtual void [Print](#) (std::ostream &) const

Static Public Member Functions

- static [STATES](#) [GetSTATES](#) (const char *state)
- static const char * [GetSTATESString](#) ([STATES](#) state)
- static [VIEWType](#) [GetVIEWType](#) (const char *type)
- static const char * [GetVIEWTypeString](#) ([VIEWType](#) type)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

12.302.1 Detailed Description

This class defines a SURFACE IE.

This members are taken from required surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.302.2 Member Enumeration Documentation

12.302.2.1 STATES

enum [gdcm::Surface::STATES](#)

Enumerator

NO	
YES	

UNKNOWN	
STATES_END	

12.302.2.2 VIEWType

enum [gdcmm::Surface::VIEWType](#)

Enumeration for Recommended Presentation [Type](#).

See also

Tag(0x0066, 0x000D) and PS 3.3 C.27.1.1.3

Enumerator

SURFACE	
WIREFRAME	
POINTS	
VIEWType_END	

12.302.3 Constructor & Destructor Documentation

12.302.3.1 Surface()

[gdcmm::Surface::Surface](#) ()

12.302.3.2 ~Surface()

[gdcmm::Surface::~~Surface](#) () [override]

12.302.4 Member Function Documentation

12.302.4.1 GetAlgorithmFamily() [1/2]

[SegmentHelper::BasicCodedEntry](#) & [gdcmm::Surface::GetAlgorithmFamily](#) ()

12.302.4.2 GetAlgorithmFamily() [2/2]

[SegmentHelper::BasicCodedEntry](#) const & [gdcmm::Surface::GetAlgorithmFamily](#) () const

12.302.4.3 GetAlgorithmName()

```
const char * gdcm::Surface::GetAlgorithmName () const
```

12.302.4.4 GetAlgorithmVersion()

```
const char * gdcm::Surface::GetAlgorithmVersion () const
```

12.302.4.5 GetAxisOfRotation()

```
const float * gdcm::Surface::GetAxisOfRotation () const
```

Note

Pointer is null if undefined

12.302.4.6 GetCenterOfRotation()

```
const float * gdcm::Surface::GetCenterOfRotation () const
```

Note

Pointer is null if undefined

12.302.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume () const
```

12.302.4.8 GetManifold()

```
STATES gdcm::Surface::GetManifold () const
```

12.302.4.9 GetMaximumPointDistance()

```
float gdcm::Surface::GetMaximumPointDistance () const
```

12.302.4.10 GetMeanPointDistance()

```
float gdcm::Surface::GetMeanPointDistance () const
```

12.302.4.11 GetMeshPrimitive() [1/2]

[MeshPrimitive](#) & gdcm::Surface::GetMeshPrimitive ()

12.302.4.12 GetMeshPrimitive() [2/2]

[MeshPrimitive](#) const & gdcm::Surface::GetMeshPrimitive () const

12.302.4.13 GetNumberOfSurfacePoints()

unsigned long gdcm::Surface::GetNumberOfSurfacePoints () const

12.302.4.14 GetNumberOfVectors()

unsigned long gdcm::Surface::GetNumberOfVectors () const

12.302.4.15 GetPointCoordinatesData() [1/2]

[DataElement](#) & gdcm::Surface::GetPointCoordinatesData ()

12.302.4.16 GetPointCoordinatesData() [2/2]

const [DataElement](#) & gdcm::Surface::GetPointCoordinatesData () const

12.302.4.17 GetPointPositionAccuracy()

const float * gdcm::Surface::GetPointPositionAccuracy () const

Note

Pointer is null if undefined

12.302.4.18 GetPointsBoundingBoxCoordinates()

const float * gdcm::Surface::GetPointsBoundingBoxCoordinates () const

Note

Pointer is null if undefined

12.302.4.19 GetProcessingAlgorithm() [1/2]

[SegmentHelper::BasicCodedEntry](#) & gdcm::Surface::GetProcessingAlgorithm ()

12.302.4.20 GetProcessingAlgorithm() [2/2]

[SegmentHelper::BasicCodedEntry](#) const & gdcm::Surface::GetProcessingAlgorithm () const

12.302.4.21 GetRecommendedDisplayCIELabValue() [1/2]

const unsigned short * gdcm::Surface::GetRecommendedDisplayCIELabValue () const

12.302.4.22 GetRecommendedDisplayCIELabValue() [2/2]

unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
const unsigned int idx) const

12.302.4.23 GetRecommendedDisplayGrayscaleValue()

unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue () const

12.302.4.24 GetRecommendedPresentationOpacity()

float gdcm::Surface::GetRecommendedPresentationOpacity () const

12.302.4.25 GetRecommendedPresentationType()

[VIEWType](#) gdcm::Surface::GetRecommendedPresentationType () const

12.302.4.26 GetSTATES()

[STATES](#) gdcm::Surface::GetSTATES (
const char * state) [static]

12.302.4.27 GetSTATESString()

const char * gdcm::Surface::GetSTATESString (
[STATES](#) state) [static]

12.302.4.28 GetSurfaceComments()

```
const char * gdcmm::Surface::GetSurfaceComments () const
```

12.302.4.29 GetSurfaceNumber()

```
unsigned long gdcmm::Surface::GetSurfaceNumber () const
```

12.302.4.30 GetSurfaceProcessing()

```
bool gdcmm::Surface::GetSurfaceProcessing () const
```

12.302.4.31 GetSurfaceProcessingDescription()

```
const char * gdcmm::Surface::GetSurfaceProcessingDescription () const
```

12.302.4.32 GetSurfaceProcessingRatio()

```
float gdcmm::Surface::GetSurfaceProcessingRatio () const
```

12.302.4.33 GetVectorAccuracy()

```
const float * gdcmm::Surface::GetVectorAccuracy () const
```

12.302.4.34 GetVectorCoordinateData() [1/2]

```
DataElement & gdcmm::Surface::GetVectorCoordinateData ()
```

12.302.4.35 GetVectorCoordinateData() [2/2]

```
const DataElement & gdcmm::Surface::GetVectorCoordinateData () const
```

12.302.4.36 GetVectorDimensionality()

```
unsigned short gdcmm::Surface::GetVectorDimensionality () const
```

12.302.4.37 GetVIEWType()

```
VIEWType gdcmm::Surface::GetVIEWType (  
    const char * type) [static]
```

12.302.4.38 GetVIEWTypeString()

```
const char * gdcm::Surface::GetVIEWTypeString (
    VIEWType type) [static]
```

12.302.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
    SegmentHelper::BasicCodedEntry const & BSE)
```

12.302.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
    const char * str)
```

12.302.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
    const char * str)
```

12.302.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
    const float * axis)
```

12.302.4.43 SetCenterOfRotation()

```
void gdcm::Surface::SetCenterOfRotation (
    const float * center)
```

12.302.4.44 SetFiniteVolume()

```
void gdcm::Surface::SetFiniteVolume (
    STATES state)
```

12.302.4.45 SetManifold()

```
void gdcm::Surface::SetManifold (
    STATES state)
```


12.302.4.46 SetMaximumPointDistance()

```
void gdcmm::Surface::SetMaximumPointDistance (  
    float maximum)
```

12.302.4.47 SetMeanPointDistance()

```
void gdcmm::Surface::SetMeanPointDistance (  
    float average)
```

12.302.4.48 SetMeshPrimitive()

```
void gdcmm::Surface::SetMeshPrimitive (  
    MeshPrimitive const & mp)
```

References [gdcmm::Object::SmartPointer](#).

12.302.4.49 SetNumberOfSurfacePoints()

```
void gdcmm::Surface::SetNumberOfSurfacePoints (  
    const unsigned long nb)
```

12.302.4.50 SetNumberOfVectors()

```
void gdcmm::Surface::SetNumberOfVectors (  
    const unsigned long nb)
```

12.302.4.51 SetPointCoordinatesData()

```
void gdcmm::Surface::SetPointCoordinatesData (  
    DataElement const & de)
```

12.302.4.52 SetPointPositionAccuracy()

```
void gdcmm::Surface::SetPointPositionAccuracy (  
    const float * accuracies)
```

12.302.4.53 SetPointsBoundingBoxCoordinates()

```
void gdcmm::Surface::SetPointsBoundingBoxCoordinates (  
    const float * coordinates)
```

12.302.4.54 SetProcessingAlgorithm()

```
void gdcmm::Surface::SetProcessingAlgorithm (  
    SegmentHelper::BasicCodedEntry const & BSE)
```

12.302.4.55 SetRecommendedDisplayCIELabValue() [1/3]

```
void gdcmm::Surface::SetRecommendedDisplayCIELabValue (  
    const std::vector< unsigned short > & vl)
```

12.302.4.56 SetRecommendedDisplayCIELabValue() [2/3]

```
void gdcmm::Surface::SetRecommendedDisplayCIELabValue (  
    const unsigned short vl,  
    const unsigned int idx = 0)
```

12.302.4.57 SetRecommendedDisplayCIELabValue() [3/3]

```
void gdcmm::Surface::SetRecommendedDisplayCIELabValue (  
    const unsigned short vl[3])
```

12.302.4.58 SetRecommendedDisplayGrayscaleValue()

```
void gdcmm::Surface::SetRecommendedDisplayGrayscaleValue (  
    const unsigned short vl)
```

12.302.4.59 SetRecommendedPresentationOpacity()

```
void gdcmm::Surface::SetRecommendedPresentationOpacity (  
    const float opacity)
```

12.302.4.60 SetRecommendedPresentationType()

```
void gdcmm::Surface::SetRecommendedPresentationType (  
    VIEWType type)
```

12.302.4.61 SetSurfaceComments()

```
void gdcmm::Surface::SetSurfaceComments (  
    const char * comment)
```

12.302.4.62 SetSurfaceNumber()

```
void gdcmm::Surface::SetSurfaceNumber (
    const unsigned long nb)
```

12.302.4.63 SetSurfaceProcessing()

```
void gdcmm::Surface::SetSurfaceProcessing (
    bool b)
```

12.302.4.64 SetSurfaceProcessingDescription()

```
void gdcmm::Surface::SetSurfaceProcessingDescription (
    const char * description)
```

12.302.4.65 SetSurfaceProcessingRatio()

```
void gdcmm::Surface::SetSurfaceProcessingRatio (
    const float ratio)
```

12.302.4.66 SetVectorAccuracy()

```
void gdcmm::Surface::SetVectorAccuracy (
    const float * accuracy)
```

12.302.4.67 SetVectorCoordinateData()

```
void gdcmm::Surface::SetVectorCoordinateData (
    DataElement const & de)
```

12.302.4.68 SetVectorDimensionality()

```
void gdcmm::Surface::SetVectorDimensionality (
    const unsigned short dim)
```

The documentation for this class was generated from the following file:

- [gdcmmSurface.h](#)

12.303 gdcm::SurfaceHelper Class Reference

[SurfaceHelper](#).

```
#include <gdcmSurfaceHelper.h>
```

Public Types

- typedef std::vector< unsigned short > [ColorArray](#)

Static Public Member Functions

- template<typename U>
static std::vector< float > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U rangeMax=255)
Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T, typename U>
static std::vector< T > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U rangeMax=255)
Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T, typename U>
static [ColorArray](#) [RGBToRecommendedDisplayCIELab](#) (const std::vector< T > &RGB, const U rangeMax=255)
Convert a RGB color into DICOM CIE-Lab (ready to write).
- template<typename T, typename U>
static unsigned short [RGBToRecommendedDisplayGrayscale](#) (const std::vector< T > &RGB, const U rangeMax=255)
Convert a RGB color into DICOM grayscale (ready to write).

12.303.1 Detailed Description

[SurfaceHelper](#).

Helper class for [Surface](#) object

12.303.2 Member Typedef Documentation

12.303.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

12.303.3 Member Function Documentation

12.303.3.1 RecommendedDisplayCIELabToRGB() [1/2]

```
template<typename U>
std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
    const ColorArray & CIELab,
    const U rangeMax = 255) [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

U	Type of rangeMax value.
---	---

References [RecommendedDisplayCIELabToRGB\(\)](#).

12.303.3.2 RecommendedDisplayCIELabToRGB() [2/2]

```
template<typename T, typename U>
std::vector< T > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
    const ColorArray & CIELab,
    const U rangeMax = 255) [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of CIELab components.
U	Type of rangeMax value.

References [gdcm_assert](#).

Referenced by [RecommendedDisplayCIELabToRGB\(\)](#).

12.303.3.3 RGBToRecommendedDisplayCIELab()

```
template<typename T, typename U>
SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (
    const std::vector< T > & RGB,
    const U rangeMax = 255) [static]
```

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

References [gdcm_assert](#).

12.303.3.4 RGBToRecommendedDisplayGrayscale()

```
template<typename T, typename U>
unsigned short gdcm::SurfaceHelper::RGBToRecommendedDisplayGrayscale (
    const std::vector< T > & RGB,
    const U rangeMax = 255) [static]
```

Convert a RGB color into DICOM grayscale (ready to write).

See also

PS 3.3 C.27.1 tag(0062,000C)

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

References [gdcm_assert](#).

The documentation for this class was generated from the following file:

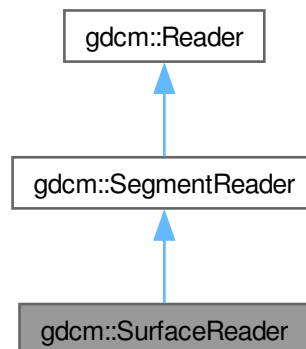
- [gdcmSurfaceHelper.h](#)

12.304 gdcm::SurfaceReader Class Reference

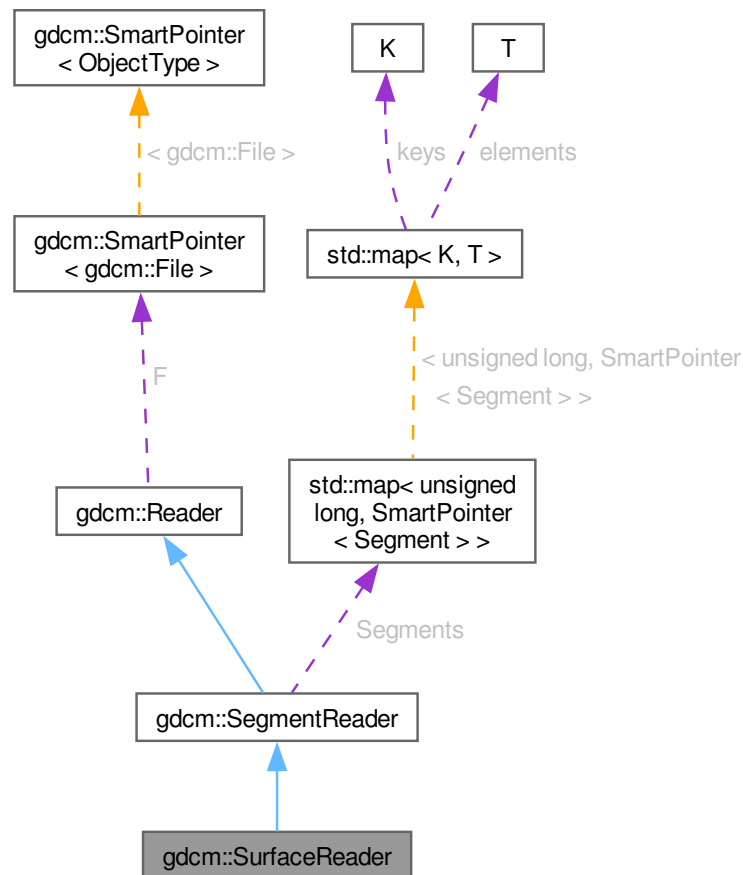
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for `gdcm::SurfaceReader`:



Public Member Functions

- [SurfaceReader](#) ()
 - [~SurfaceReader](#) () override
 - unsigned long [GetNumberOfSurfaces](#) () const
 - bool [Read](#) () override
- Read.

Public Member Functions inherited from [gdcm::SegmentReader](#)

- [SegmentReader](#) ()
- [~SegmentReader](#) () override
- [SegmentVector](#) [GetSegments](#) ()
- [SegmentVector](#) [GetSegments](#) () const

Public Member Functions inherited from [gdcm::Reader](#)

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- [File](#) & [GetFile](#) ()
Set/Get [File](#).
- const [File](#) & [GetFile](#) () const
Set/Get [File](#).
- size_t [GetStreamCurrentPosition](#) () const
- bool [ReadSelectedPrivateTags](#) (std::set< [PrivateTag](#) > const &ptags, bool readvalues=true)
Will only read the specified selected private tags.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags, bool readvalues=true)
Will only read the specified selected tags.
- bool [ReadUpToTag](#) (const [Tag](#) &tag, std::set< [Tag](#) > const &skiptags=std::set< [Tag](#) >())
- void [SetFile](#) ([File](#) &file)
Set/Get [File](#).
- void [SetFileName](#) (const char *filename_native)
- void [SetStream](#) (std::istream &input_stream)
Set the open-ed stream directly.

Protected Member Functions

- bool [ReadPointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, const [DataSet](#) &surfaceDS)
- bool [ReadSurface](#) (const [Item](#) &surfaceItem, const unsigned long idx)
- bool [ReadSurfaces](#) ()

Protected Member Functions inherited from [gdcm::SegmentReader](#)

- bool [ReadSegment](#) (const [Item](#) &segmentItem, const unsigned int idx)
- bool [ReadSegments](#) ()

Protected Member Functions inherited from [gdcm::Reader](#)

- std::istream * [GetStreamPtr](#) () const
- bool [ReadDataSet](#) ()
- bool [ReadMetaInformation](#) ()
- bool [ReadPreamble](#) ()

Additional Inherited Members

Public Types inherited from [gdcm::SegmentReader](#)

- typedef std::vector< [SmartPointer](#)< [Segment](#) > > [SegmentVector](#)

Protected Types inherited from [gdcm::SegmentReader](#)

- typedef std::map< unsigned long, [SmartPointer< Segment >](#) > [SegmentMap](#)

Protected Attributes inherited from [gdcm::SegmentReader](#)

- [SegmentMap Segments](#)

Protected Attributes inherited from [gdcm::Reader](#)

- [SmartPointer< File >](#) [F](#)

12.304.1 Detailed Description

This class defines a SURFACE IE reader.

It reads surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.304.2 Constructor & Destructor Documentation

12.304.2.1 SurfaceReader()

`gdcm::SurfaceReader::SurfaceReader ()`

12.304.2.2 ~SurfaceReader()

`gdcm::SurfaceReader::~SurfaceReader ()` [override]

12.304.3 Member Function Documentation

12.304.3.1 GetNumberOfSurfaces()

`unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces () const`

12.304.3.2 Read()

```
bool gdcm::SurfaceReader::Read () [override], [virtual]
```

Read.

Reimplemented from [gdcm::SegmentReader](#).

12.304.3.3 ReadPointMacro()

```
bool gdcm::SurfaceReader::ReadPointMacro (  
    SmartPointer< Surface > surface,  
    const DataSet & surfaceDS) [protected]
```

12.304.3.4 ReadSurface()

```
bool gdcm::SurfaceReader::ReadSurface (  
    const Item & surfaceItem,  
    const unsigned long idx) [protected]
```

12.304.3.5 ReadSurfaces()

```
bool gdcm::SurfaceReader::ReadSurfaces () [protected]
```

The documentation for this class was generated from the following file:

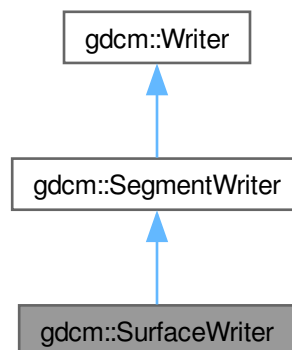
- [gdcmSurfaceReader.h](#)

12.305 gdcm::SurfaceWriter Class Reference

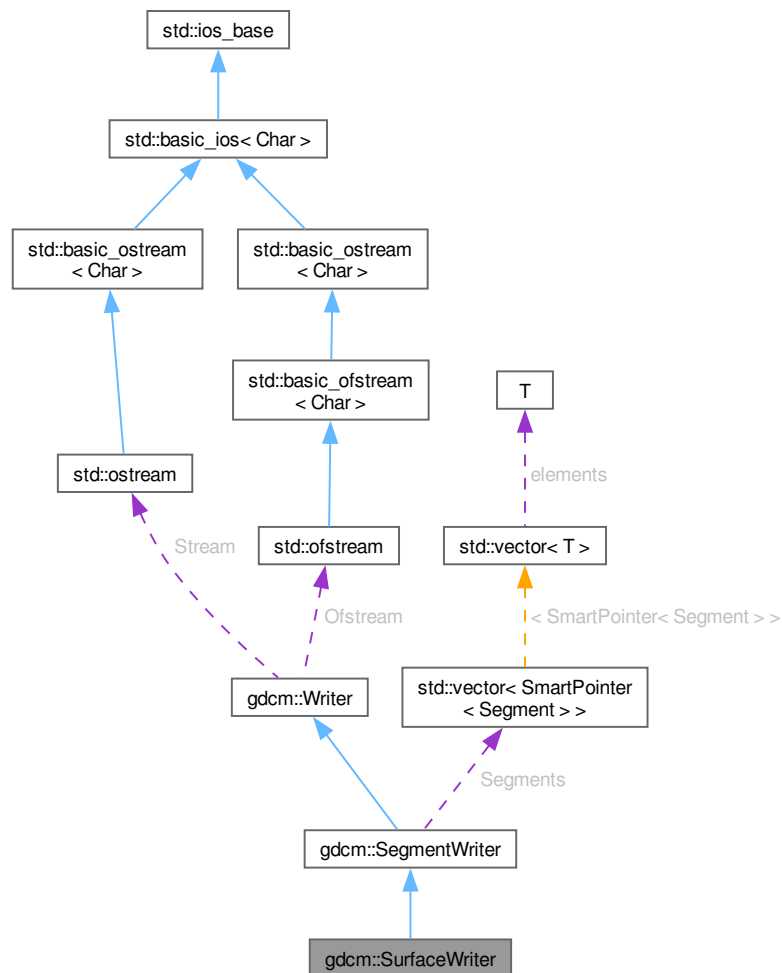
This class defines a SURFACE IE writer.

```
#include <gdcmSurfaceWriter.h>
```

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for `gdc::SurfaceWriter`:



Public Member Functions

- [SurfaceWriter](#) ()
 - [~SurfaceWriter](#) () override
 - unsigned long [GetNumberOfSurfaces](#) ()
 - void [SetNumberOfSurfaces](#) (const unsigned long nb)
 - bool [Write](#) () override
- Write.

Public Member Functions inherited from [gdc::SegmentWriter](#)

- [SegmentWriter](#) ()

- [~SegmentWriter](#) () override
- void [AddSegment](#) ([SmartPointer](#)< [Segment](#) > segment)
- unsigned int [GetNumberOfSegments](#) () const
- [SmartPointer](#)< [Segment](#) > [GetSegment](#) (const unsigned int idx=0) const
- [SegmentVector](#) & [GetSegments](#) ()
- const [SegmentVector](#) & [GetSegments](#) () const
- void [SetNumberOfSegments](#) (const unsigned int size)
- void [SetSegments](#) ([SegmentVector](#) &segments)

Public Member Functions inherited from [gdcm::Writer](#)

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)
Undocumented function, do not use (= leave default).
- void [SetFile](#) (const [File](#) &f)
Set/Get the DICOM file ([DataSet](#) + Header).
- void [SetFileName](#) (const char *filename_native)
Set the filename of DICOM file to write:
- void [SetStream](#) (std::ostream &output_stream)
Set user ostream buffer.

Protected Member Functions

- void [ComputeNumberOfSurfaces](#) ()
- bool [PrepareWrite](#) ()
- bool [PrepareWritePointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, [DataSet](#) &surfaceDS, const [TransferSyntax](#) &ts)

Protected Member Functions inherited from [gdcm::SegmentWriter](#)

- bool [PrepareWrite](#) ()

Protected Member Functions inherited from [gdcm::Writer](#)

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes

- unsigned long [NumberOfSurfaces](#)

Protected Attributes inherited from [gdcm::SegmentWriter](#)

- [SegmentVector](#) [Segments](#)

Protected Attributes inherited from [gdcm::Writer](#)

- `std::ofstream` * [Ofstream](#)
- `std::ostream` * [Stream](#)

Additional Inherited Members

Public Types inherited from [gdcm::SegmentWriter](#)

- `typedef std::vector< SmartPointer< Segment > > SegmentVector`

12.305.1 Detailed Description

This class defines a SURFACE IE writer.

It writes surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.305.2 Constructor & Destructor Documentation

12.305.2.1 `SurfaceWriter()`

`gdcm::SurfaceWriter::SurfaceWriter ()`

12.305.2.2 `~SurfaceWriter()`

`gdcm::SurfaceWriter::~SurfaceWriter ()` [override]

12.305.3 Member Function Documentation

12.305.3.1 `ComputeNumberOfSurfaces()`

`void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ()` [protected]

12.305.3.2 GetNumberOfSurfaces()

unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ()

12.305.3.3 PrepareWrite()

bool gdcm::SurfaceWriter::PrepareWrite () [protected]

12.305.3.4 PrepareWritePointMacro()

bool gdcm::SurfaceWriter::PrepareWritePointMacro (
 [SmartPointer< Surface >](#) surface,
 [DataSet](#) & surfaceDS,
 const [TransferSyntax](#) & ts) [protected]

12.305.3.5 SetNumberOfSurfaces()

void gdcm::SurfaceWriter::SetNumberOfSurfaces (
 const unsigned long nb)

12.305.3.6 Write()

bool gdcm::SurfaceWriter::Write () [override], [virtual]

Write.

Reimplemented from [gdcm::SegmentWriter](#).

12.305.4 Member Data Documentation

12.305.4.1 NumberOfSurfaces

unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]

The documentation for this class was generated from the following file:

- [gdcmSurfaceWriter.h](#)

12.306 gdcm::SwapCode Class Reference

[SwapCode](#) representation.

```
#include <gdcmSwapCode.h>
```

Public Types

- enum [SwapCodeType](#) {
[Unknown](#) = 0 ,
[LittleEndian](#) = 1234 ,
[BigEndian](#) = 4321 ,
[BadLittleEndian](#) = 3412 ,
[BadBigEndian](#) = 2143 }

Public Member Functions

- [SwapCode](#) ([SwapCodeType](#) sc=[Unknown](#))
- [operator SwapCode::SwapCodeType](#) () const

Static Public Member Functions

- static const char * [GetSwapCodeString](#) ([SwapCode](#) const &sc)

Static Protected Member Functions

- static int [GetIndex](#) ([SwapCode](#) const &sc)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [SwapCode](#) &sc)

12.306.1 Detailed Description

[SwapCode](#) representation.

Examples

[TestByteSwap.cxx](#).

12.306.2 Member Enumeration Documentation

12.306.2.1 SwapCodeType

enum [gdcmm::SwapCode::SwapCodeType](#)

Enumerator

Unknown	
-------------------------	--

LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

12.306.3 Constructor & Destructor Documentation

12.306.3.1 SwapCode()

gdcm::SwapCode::SwapCode (
[SwapCodeType](#) sc = [Unknown](#)) [inline]

References [Unknown](#).

Referenced by [GetIndex\(\)](#), [GetSwapCodeString\(\)](#), and [operator<<](#).

12.306.4 Member Function Documentation

12.306.4.1 GetIndex()

int gdcm::SwapCode::GetIndex (
[SwapCode](#) const & sc) [static], [protected]

References [SwapCode\(\)](#).

12.306.4.2 GetSwapCodeString()

const char * gdcm::SwapCode::GetSwapCodeString (
[SwapCode](#) const & sc) [static]

References [SwapCode\(\)](#), and [operator<<](#).

Referenced by [operator<<](#).

12.306.4.3 operator SwapCode::SwapCodeType()

gdcm::SwapCode::operator [SwapCode::SwapCodeType](#) () const [inline]

12.306.5 Friends And Related Symbol Documentation

12.306.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const SwapCode & sc) [friend]
```

References [SwapCode\(\)](#), and [GetSwapCodeString\(\)](#).

Referenced by [GetSwapCodeString\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmswapcode.h](#)

12.307 gdcmswapperdoop Class Reference

```
#include <gdcmswapper.h>
```

Static Public Member Functions

- `template<typename T>
static T Swap (T val)`
- `template<typename T>
static void SwapArray (T *array, size_t n)`

12.307.1 Member Function Documentation

12.307.1.1 Swap()

```
template<typename T>  
T gdcmswapperdoop::Swap (  
    T val) [static]
```

Referenced by [gdcmswapperdoop::Read\(\)](#), and [SwapArray\(\)](#).

12.307.1.2 SwapArray()

```
template<typename T>  
void gdcmswapperdoop::SwapArray (  
    T * array,  
    size_t n) [inline], [static]
```

References [Swap\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmswapper.h](#)

12.308 gdcm::SwapperNoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- `template<typename T>`
`static T Swap (T val)`
- `template<typename T>`
`static void SwapArray (T *, size_t)`

12.308.1 Detailed Description

Examples

[DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DumpToshibaDTI2.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

12.308.2 Member Function Documentation

12.308.2.1 [Swap\(\)](#)

```
template<typename T>  
T gdcm::SwapperNoOp::Swap (  
    T val) [inline], [static]
```

Referenced by [gdcm::EncodingImplementation< VR::VRBINARY >::Write\(\)](#).

12.308.2.2 [SwapArray\(\)](#)

```
template<typename T>  
void gdcm::SwapperNoOp::SwapArray (  
    T *,  
    size_t ) [inline], [static]
```

Referenced by [gdcm::EncodingImplementation< VR::VRBINARY >::Read\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmSwapper.h](#)

12.309 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

Static Public Member Functions

- static std::wstring [ConvertToUNC](#) (const char *utf8path)
- static bool [DeleteDirectory](#) (const char *source)
 - remove a directory named source
- static size_t [EncodeBytes](#) (char *out, const unsigned char *data, int size)
- static bool [FileExists](#) (const char *filename)
 - Check whether the specified file exist on the system.
- static bool [FileIsDirectory](#) (const char *name)
 - Check whether the file specified is a directory:
- static bool [FileIsSymlink](#) (const char *name)
 - Check whether name is a symlink.
- static size_t [FileSize](#) (const char *filename)
- static time_t [FileTime](#) (const char *filename)
- static bool [FormatDateTime](#) (char date[22], time_t t, long milliseconds=0)
- static bool [GetCurrentDateTime](#) (char date[22])
- static const char * [GetCurrentModuleFileName](#) ()
- static const char * [GetCurrentProcessFileName](#) ()
- static const char * [GetCurrentResourcesDirectory](#) ()
- static const char * [GetCWD](#) ()
- static bool [GetHostName](#) (char hostname[255])
- static const char * [GetLastError](#) ()
 - Return the last error.
- static const char * [GetLocaleCharset](#) ()
 - return locale charmap
- static const char * [GetTimezoneOffsetFromUTC](#) ()
- static bool [MakeDirectory](#) (const char *path)
 - Create a directory name path.
- static bool [ParseDateTime](#) (time_t &timep, const char date[22])
 - Parse a date stored as ASCII text into a time_t structured (discard millisecond if any).
- static bool [ParseDateTime](#) (time_t &timep, long &milliseconds, const char date[22])
- static bool [RemoveFile](#) (const char *source)
 - remove a file named source
- static int [StrCaseCmp](#) (const char *s1, const char *s2)
 - consistent func for C99 spec of strcasecmp/strncasecmp
- static int [StrNCaseCmp](#) (const char *s1, const char *s2, size_t n)
- static char * [StrSep](#) (char **stringp, const char *delim)
- static char * [StrTokR](#) (char *ptr, const char *sep, char **end)
 - strtok_r

Static Protected Member Functions

- static bool [GetPermissions](#) (const char *file, unsigned short &mode)
 - NOT THREAD SAFE.
- static bool [SetPermissions](#) (const char *file, unsigned short mode)

12.309.1 Detailed Description

Class to do system operation.

OS independent functionalities

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [Cleaner.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [DumpCSA.cs](#), [ExplicitLittleEndian.cs](#), [ExtractEncapsulatedFile.cs](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [ExtractOneFrame.cs](#), [FileAnonymize.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FileStreaming.cs](#), [GetArray.cs](#), [MetaImageMD5Activiz.cs](#), [MpegVideoInfo.cs](#), [ReformatFile.cs](#), [RescaleImage.cs](#), [ScanDirectory.cs](#), [SimplePrint.cs](#), and [StandardizeFiles.cs](#).

12.309.2 Member Function Documentation

12.309.2.1 ConvertToUNC()

```
std::wstring gdcm::System::ConvertToUNC (  
    const char * utf8path) [static]
```

When needed convert a PATH into a UNC equivalent. This allow transparent support for path longer than MAX_PATH. Only on _MSC_VER compiler, return empty string otherwise.

12.309.2.2 DeleteDirectory()

```
bool gdcm::System::DeleteDirectory (  
    const char * source) [static]
```

remove a directory named source

12.309.2.3 EncodeBytes()

```
size_t gdcm::System::EncodeBytes (  
    char * out,  
    const unsigned char * data,  
    int size) [static]
```

Used internally by the [UIDGenerator](#) class to convert a uuid tape to a DICOM [VR:UI](#) type

12.309.2.4 FileExists()

```
bool gdcm::System::FileExists (  
    const char * filename) [static]
```

Check whether the specified file exist on the system.

Examples

[DumpVisusChange.cxx](#), [EncapsulateFileInRawData.cxx](#), [MagnifyFile.cxx](#), and [gdcmorthoplanes.cxx](#).

12.309.2.5 FileIsDirectory()

```
bool gdcmm::System::FileIsDirectory (  
    const char * name) [static]
```

Check whether the file specified is a directory:

Examples

[DumpVisusChange.cxx](#), [gdcmmorthoplanes.cxx](#), and [threadgdcmm.cxx](#).

12.309.2.6 FileIsSymlink()

```
bool gdcmm::System::FileIsSymlink (  
    const char * name) [static]
```

Check whether name is a symlink.

12.309.2.7 FileSize()

```
size_t gdcmm::System::FileSize (  
    const char * filename) [static]
```

Return the filesize. 0 if file does not exist.

Warning

you need to use FileExists to differentiate between empty file and missing file.

for very large size file and on system where size_t is not appropriate to store off_t value the function will return 0.

Examples

[CheckBigEndianBug.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [EncapsulateFileInRawData.cxx](#), and [SimpleScanner.cxx](#).

12.309.2.8 FileTime()

```
time_t gdcmm::System::FileTime (  
    const char * filename) [static]
```

Return the time of last modification of file 0 if the file does not exist

12.309.2.9 FormatDateTime()

```
bool gdcm::System::FormatDateTime (
    char date[22],
    time_t t,
    long milliseconds = 0) [static]
```

format as ASCII text a time_t with milliseconds See [VR::DT](#) from DICOM PS 3.5 milliseconds is in the range [0, 999999]

12.309.2.10 GetCurrentDateTime()

```
bool gdcm::System::GetCurrentDateTime (
    char date[22]) [static]
```

Return the current data time, and format it as ASCII text. This is simply a call to `gettimeofday + FormatDateTime`, since WIN32 do not have an implementation for `gettimeofday`, this is more portable. The call `time(0)` is not precise for our resolution

Examples

[TemplateEmptyImage.cxx](#).

12.309.2.11 GetCurrentModuleFileName()

```
const char * gdcm::System::GetCurrentModuleFileName () [static]
```

Return the directory the current module is located: NOT THREAD SAFE

12.309.2.12 GetCurrentProcessFileName()

```
const char * gdcm::System::GetCurrentProcessFileName () [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

12.309.2.13 GetCurrentResourcesDirectory()

```
const char * gdcm::System::GetCurrentResourcesDirectory () [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

12.309.2.14 GetCWD()

```
const char * gdcmm::System::GetCWD () [static]
```

Return current working directory Warning: if current working path is too long (>2048 bytes) the call will fail and call will return NULL NOT THREAD SAFE

12.309.2.15 GetHostName()

```
bool gdcmm::System::GetHostName (  
    char hostname[255]) [static]
```

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

12.309.2.16 GetLastSystemError()

```
const char * gdcmm::System::GetLastSystemError () [static]
```

Return the last error.

12.309.2.17 GetLocaleCharset()

```
const char * gdcmm::System::GetLocaleCharset () [static]
```

return locale charmap

12.309.2.18 GetPermissions()

```
bool gdcmm::System::GetPermissions (  
    const char * file,  
    unsigned short & mode) [static], [protected]
```

NOT THREAD SAFE.

12.309.2.19 GetTimezoneOffsetFromUTC()

```
const char * gdcmm::System::GetTimezoneOffsetFromUTC () [static]
```

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

12.309.2.20 MakeDirectory()

```
bool gdcm::System::MakeDirectory (  
    const char * path) [static]
```

Create a directory name path.

12.309.2.21 ParseDateTime() [1/2]

```
bool gdcm::System::ParseDateTime (  
    time_t & timep,  
    const char date[22]) [static]
```

Parse a date stored as ASCII text into a time_t structured (discard millisecond if any).

12.309.2.22 ParseDateTime() [2/2]

```
bool gdcm::System::ParseDateTime (  
    time_t & timep,  
    long & milliseconds,  
    const char date[22]) [static]
```

Parse a date stored as ASCII text into a time_t structured and millisecond

See also

[FormatDateTime](#)

12.309.2.23 RemoveFile()

```
bool gdcm::System::RemoveFile (  
    const char * source) [static]
```

remove a file named source

12.309.2.24 SetPermissions()

```
bool gdcm::System::SetPermissions (  
    const char * file,  
    unsigned short mode) [static], [protected]
```

12.309.2.25 StrCaseCmp()

```
int gdcm::System::StrCaseCmp (  
    const char * s1,  
    const char * s2) [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

12.309.2.26 StrNCaseCmp()

```
int gdcM::System::StrNCaseCmp (
    const char * s1,
    const char * s2,
    size_t n) [static]
```

Precondition

n != 0

12.309.2.27 StrSep()

```
char * gdcM::System::StrSep (
    char ** stringp,
    const char * delim) [static]
```

strsep param stringp is passed by pointer, it may be modified, you'll need to make a copy, in case you want to free the memory pointed at

12.309.2.28 StrTokR()

```
char * gdcM::System::StrTokR (
    char * ptr,
    const char * sep,
    char ** end) [static]
```

strtok_r

The documentation for this class was generated from the following file:

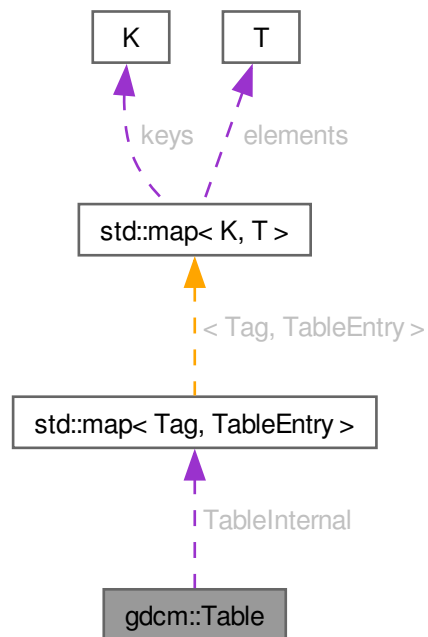
- [gdcMSystem.h](#)

12.310 gdcM::Table Class Reference

[Table](#).

```
#include <gdcMTable.h>
```

Collaboration diagram for gdcm::Table:



Public Types

- typedef std::map< [Tag](#), [TableEntry](#) > [MapTableEntry](#)

Public Member Functions

- [Table](#) ()=default
- [Table](#) (const [Table](#) &_val)=delete
- [~Table](#) ()=default
- const [TableEntry](#) & [GetTableEntry](#) (const [Tag](#) &tag) const
- void [InsertEntry](#) ([Tag](#) const &tag, [TableEntry](#) const &te)
- [Table](#) & [operator=](#) (const [Table](#) &_val)=delete

Public Attributes

- [MapTableEntry](#) [TableInternal](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Table](#) &_val)

12.310.1 Detailed Description

[Table](#).

12.310.2 Member Typedef Documentation

12.310.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcmm::Table::MapTableEntry
```

12.310.3 Constructor & Destructor Documentation

12.310.3.1 Table() [1/2]

```
gdcmm::Table::Table () [default]
```

Referenced by [Table\(\)](#), [operator<<](#), and [operator=\(\)](#).

12.310.3.2 ~Table()

```
gdcmm::Table::~~Table () [default]
```

12.310.3.3 Table() [2/2]

```
gdcmm::Table::Table (  
    const Table & _val) [delete]
```

References [Table\(\)](#).

12.310.4 Member Function Documentation

12.310.4.1 GetTableEntry()

```
const TableEntry & gdcmm::Table::GetTableEntry (  
    const Tag & tag) const [inline]
```

References [gdcmm_assert](#), [GetTableEntry\(\)](#), and [TableInternal](#).

Referenced by [GetTableEntry\(\)](#).

12.310.4.2 InsertEntry()

```
void gdcM::Table::InsertEntry (
    Tag const & tag,
    TableEntry const & te) [inline]
```

References [gdcM_assert](#), and [TableInternal](#).

12.310.4.3 operator=()

```
Table & gdcM::Table::operator= (
    const Table & _val) [delete]
```

References [Table\(\)](#).

12.310.5 Friends And Related Symbol Documentation

12.310.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & _os,
    const Table & _val) [friend]
```

References [Table\(\)](#).

12.310.6 Member Data Documentation

12.310.6.1 TableInternal

[MapTableEntry](#) gdcM::Table::TableInternal

Referenced by [GetTableEntry\(\)](#), and [InsertEntry\(\)](#).

The documentation for this class was generated from the following file:

- [gdcMTable.h](#)

12.311 gdcM::TableEntry Class Reference

[TableEntry](#).

```
#include <gdcMTableEntry.h>
```

Public Member Functions

- [TableEntry](#) (const char *attribute=nullptr, [Type](#) const &type=[Type](#)(), const char *des=nullptr)
- [~TableEntry](#) ()=default

12.311.1 Detailed Description

[TableEntry](#).

12.311.2 Constructor & Destructor Documentation

12.311.2.1 TableEntry()

```
gdcmm::TableEntry::TableEntry (  
    const char * attribute = nullptr,  
    Type const & type = Type(),  
    const char * des = nullptr) [inline]
```

12.311.2.2 ~TableEntry()

```
gdcmm::TableEntry::~~TableEntry () [default]
```

The documentation for this class was generated from the following file:

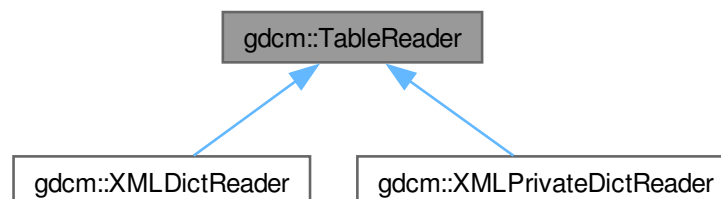
- [gdcmmTableEntry.h](#)

12.312 gdcmm::TableReader Class Reference

Class for representing a [TableReader](#).

```
#include <gdcmmTableReader.h>
```

Inheritance diagram for gdcmm::TableReader:



Public Member Functions

- [TableReader](#) ([Defs](#) &defs)
- virtual [~TableReader](#) ()=default
- virtual void [CharacterDataHandler](#) (const char *data, int length)
- virtual void [EndElement](#) (const char *name)
- const [Defs](#) & [GetDefs](#) () const
- const char * [GetFilename](#) ()
- void [HandleIOD](#) (const char **atts)
- void [HandleIODEntry](#) (const char **atts)
- void [HandleMacro](#) (const char **atts)
- void [HandleMacroEntry](#) (const char **atts)
- void [HandleMacroEntryDescription](#) (const char **atts)
- void [HandleModule](#) (const char **atts)
- void [HandleModuleEntry](#) (const char **atts)
- void [HandleModuleEntryDescription](#) (const char **atts)
- void [HandleModuleInclude](#) (const char **atts)
- int [Read](#) ()
- void [SetFilename](#) (const char *filename)
- virtual void [StartElement](#) (const char *name, const char **atts)

12.312.1 Detailed Description

Class for representing a [TableReader](#).

Note

This class is an empty shell meant to be derived

12.312.2 Constructor & Destructor Documentation

12.312.2.1 [TableReader\(\)](#)

```
gdcm::TableReader::TableReader (  
    Defs & defs) [inline]
```

12.312.2.2 [~TableReader\(\)](#)

```
virtual gdcm::TableReader::~TableReader () [virtual], [default]
```

12.312.3 Member Function Documentation

12.312.3.1 [CharacterDataHandler\(\)](#)

```
virtual void gdcm::TableReader::CharacterDataHandler (  
    const char * data,  
    int length) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

12.312.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (  
    const char * name) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

12.312.3.3 GetDefs()

```
const Defs & gdcm::TableReader::GetDefs () const [inline]
```

12.312.3.4 GetFilename()

```
const char * gdcm::TableReader::GetFilename () [inline]
```

12.312.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (  
    const char ** atts)
```

12.312.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (  
    const char ** atts)
```

12.312.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (  
    const char ** atts)
```

12.312.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (  
    const char ** atts)
```

12.312.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (  
    const char ** atts)
```


12.312.3.10 HandleModule()

```
void gdcmm::TableReader::HandleModule (  
    const char ** atts)
```

12.312.3.11 HandleModuleEntry()

```
void gdcmm::TableReader::HandleModuleEntry (  
    const char ** atts)
```

12.312.3.12 HandleModuleEntryDescription()

```
void gdcmm::TableReader::HandleModuleEntryDescription (  
    const char ** atts)
```

12.312.3.13 HandleModuleInclude()

```
void gdcmm::TableReader::HandleModuleInclude (  
    const char ** atts)
```

12.312.3.14 Read()

```
int gdcmm::TableReader::Read ()
```

12.312.3.15 SetFilename()

```
void gdcmm::TableReader::SetFilename (  
    const char * filename) [inline]
```

12.312.3.16 StartElement()

```
virtual void gdcmm::TableReader::StartElement (  
    const char * name,  
    const char ** atts) [virtual]
```

Reimplemented in [gdcmm::XMLDictReader](#), and [gdcmm::XMLPrivateDictReader](#).

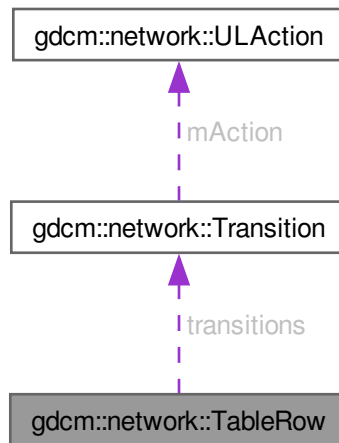
The documentation for this class was generated from the following file:

- [gdcmmTableReader.h](#)

12.313 gdcmm::network::TableRow Class Reference

```
#include <gdcmmULTransitionTable.h>
```

Collaboration diagram for gdcmm::network::TableRow:



Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

Public Attributes

- [Transition](#) * [transitions](#) [[cMaxStateID](#)]

12.313.1 Constructor & Destructor Documentation

12.313.1.1 TableRow()

```
gdcmm::network::TableRow::TableRow () [inline]
```

References [gdcmm::network::cMaxStateID](#), and [transitions](#).

12.313.1.2 ~TableRow()

gdcm::network::TableRow::~TableRow () [inline]

References [gdcm::network::cMaxStateID](#), and [transitions](#).

12.313.2 Member Data Documentation

12.313.2.1 transitions

[Transition*](#) gdcm::network::TableRow::transitions[cMaxStateID]

Referenced by [TableRow\(\)](#), and [~TableRow\(\)](#).

The documentation for this class was generated from the following file:

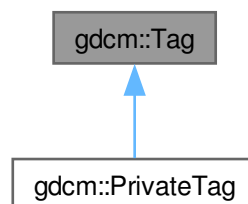
- [gdcmULTransitionTable.h](#)

12.314 gdcm::Tag Class Reference

Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

```
#include <gdcmTag.h>
```

Inheritance diagram for gdcm::Tag:



Public Member Functions

- [Tag](#) (const [Tag](#) &_val)
- [Tag](#) (uint16_t group, uint16_t element)
 Constructor with 2*uint16_t.
- [Tag](#) (uint32_t tag=0)
 Constructor with 1*uint32_t Prefer the ctor that takes two uint16_t.
- uint16_t [GetElement](#) () const
 Returns the '[Element](#) number' of the given [Tag](#).
- uint32_t [GetElementTag](#) () const
 Returns the full tag value of the given [Tag](#).
- uint16_t [GetGroup](#) () const
 Returns the 'Group number' of the given [Tag](#).
- uint32_t [GetLength](#) () const
 return the length of tag (read: size on disk)
- [Tag](#) [GetPrivateCreator](#) () const
 Return the Private Creator Data [Element](#) tag of a private data element.
- bool [IsGroupLength](#) () const
 return whether the tag correspond to a group length tag:
- bool [IsGroupXX](#) (const [Tag](#) &t) const
 e.g 6002,3000 belong to groupXX: 6000,3000
- bool [IsIllegal](#) () const
 return if the tag is considered to be an illegal tag
- bool [IsPrivate](#) () const
- bool [IsPrivateCreator](#) () const
- bool [IsPublic](#) () const
- bool [operator!=](#) (const [Tag](#) &_val) const
- bool [operator<](#) (const [Tag](#) &_val) const
- bool [operator<=](#) (const [Tag](#) &t2) const
- [Tag](#) & [operator=](#) (const [Tag](#) &_val)
- bool [operator==](#) (const [Tag](#) &_val) const
- uint16_t & [operator\[\]](#) (const unsigned int &_id)
 Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).
- const uint16_t & [operator\[\]](#) (const unsigned int &_id) const
 Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).
- std::string [PrintAsContinuousString](#) () const
- std::string [PrintAsContinuousUpperCaseString](#) () const
 Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.
- std::string [PrintAsPipeSeparatedString](#) () const
- template<typename TSwap>
 std::istream & [Read](#) (std::istream &is)
 Read a tag from binary representation.
- bool [ReadFromCommaSeparatedString](#) (const char *str)
- bool [ReadFromContinuousString](#) (const char *str)
- bool [ReadFromPipeSeparatedString](#) (const char *str)
- void [SetElement](#) (uint16_t element)
 Sets the '[Element](#) number' of the given [Tag](#).
- void [SetElementTag](#) (uint16_t group, uint16_t element)

- Sets the 'Group number' & 'Element number' of the given [Tag](#).
- void [SetElementTag](#) (uint32_t tag)
 - Sets the full tag value of the given [Tag](#).
- void [SetGroup](#) (uint16_t group)
 - Sets the 'Group number' of the given [Tag](#).
- void [SetPrivateCreator](#) ([Tag](#) const &t)
 - Set private creator:
- template<typename TSwap>
 const std::ostream & [Write](#) (std::ostream &os) const
 - Write a tag in binary rep.

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Tag](#) &_val)
- std::istream & [operator>>](#) (std::istream &_is, [Tag](#) &_val)

12.314.1 Detailed Description

Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Note

DATA ELEMENT TAG: A unique identifier for a Data [Element](#) composed of an ordered pair of numbers (a Group Number followed by an [Element](#) Number). GROUP NUMBER: The first number in the ordered pair of numbers that makes up a Data [Element Tag](#). ELEMENT NUMBER: The second number in the ordered pair of numbers that makes up a Data [Element Tag](#).

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DiscriminateVolume.cxx](#), [DumpToSQLite3.cxx](#), [DumpVisusChange.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncapsulatedFile.cs](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FileAnonymize.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [PatchFile.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ScanDirectory.cs](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), [VolumeSorter.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.314.2 Constructor & Destructor Documentation

12.314.2.1 Tag() [1/3]

```
gdcM::Tag::Tag (
    uint16_t group,
    uint16_t element) [inline]
```

Constructor with 2*uint16_t.

Referenced by [gdcM::PrivateTag::PrivateTag\(\)](#), [gdcM::PrivateTag::PrivateTag\(\)](#), [Tag\(\)](#), [GetPrivateCreator\(\)](#), [IsGroupXX\(\)](#), [gdcM::PrivateTag::operator!=\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator<<\(\)](#), [operator<=\(\)](#), [operator=\(\)](#), [gdcM::PrivateTag::operator==\(\)](#), [operator==\(\)](#), [operator>>\(\)](#), and [SetPrivateCreator\(\)](#).

12.314.2.2 Tag() [2/3]

```
gdcM::Tag::Tag (
    uint32_t tag = 0) [inline]
```

Constructor with 1*uint32_t Prefer the ctor that takes two uint16_t.

References [SetElementTag\(\)](#), and [tag](#).

12.314.2.3 Tag() [3/3]

```
gdcM::Tag::Tag (
    const Tag & _val) [inline]
```

References [Tag\(\)](#), and [tag](#).

12.314.3 Member Function Documentation

12.314.3.1 GetElement()

```
uint16_t gdcM::Tag::GetElement () const [inline]
```

Returns the 'Element number' of the given [Tag](#).

Examples

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [gdcM::PrivateTag::PrivateTag\(\)](#), [gdcM::DataSet::ComputeGroupLength\(\)](#), [GetPrivateCreator\(\)](#), [IsGroupLength\(\)](#), [IsGroupXX\(\)](#), [IsIllegal\(\)](#), [IsPrivateCreator\(\)](#), [gdcM::PrivateDict::PrintXML\(\)](#), [gdcM::SequenceOfFragments](#) and [SetPrivateCreator\(\)](#).

12.314.3.2 GetElementTag()

uint32_t gdcmm::Tag::GetElementTag () const [inline]

Returns the full tag value of the given [Tag](#).

Referenced by [gdcmm::PrivateTag::operator!=\(\)](#), [gdcmm::PrivateTag::operator!=\(\)](#), [gdcmm::PrivateTag::operator=\(\)](#), [gdcmm::PrivateTag::operator==\(\)](#), and [gdcmm::PrivateTag::operator==\(\)](#).

12.314.3.3 GetGroup()

uint16_t gdcmm::Tag::GetGroup () const [inline]

Returns the 'Group number' of the given [Tag](#).

Examples

[DuplicatePCDE.cxx](#), and [GenAllVR.cxx](#).

Referenced by [gdcmm::DataSet::ComputeGroupLength\(\)](#), [gdcmm::CommandDataSet::Insert\(\)](#), [gdcmm::DataSet::Insert\(\)](#), [gdcmm::FileMetaInformation::Insert\(\)](#), [IsGroupXX\(\)](#), [IsIllegal\(\)](#), [gdcmm::PrivateDict::PrintXML\(\)](#), [gdcmm::SequenceOfFragments](#) and [SetPrivateCreator\(\)](#).

12.314.3.4 GetLength()

uint32_t gdcmm::Tag::GetLength () const [inline]

return the length of tag (read: size on disk)

12.314.3.5 GetPrivateCreator()

[Tag](#) gdcmm::Tag::GetPrivateCreator () const [inline]

Return the Private Creator Data [Element](#) tag of a private data element.

References [Tag\(\)](#), [GetElement\(\)](#), [IsPrivate\(\)](#), [IsPrivateCreator\(\)](#), and [SetElement\(\)](#).

12.314.3.6 IsGroupLength()

bool gdcmm::Tag::IsGroupLength () const [inline]

return whether the tag correspond to a group length tag:

References [GetElement\(\)](#).

12.314.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
    const Tag & t) const    [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References [Tag\(\)](#), [GetElement\(\)](#), [GetGroup\(\)](#), and [IsPrivate\(\)](#).

12.314.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal () const    [inline]
```

return if the tag is considered to be an illegal tag

References [GetElement\(\)](#), [GetGroup\(\)](#), and [IsPrivate\(\)](#).

12.314.3.9 IsPrivate()

```
bool gdcm::Tag::IsPrivate () const    [inline]
```

PRIVATE DATA ELEMENT: Additional Data [Element](#), defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

Examples

[DuplicatePCDE.cxx](#).

References [IsPublic\(\)](#).

Referenced by [GetPrivateCreator\(\)](#), [IsGroupXX\(\)](#), [IsIllegal\(\)](#), [IsPrivateCreator\(\)](#), and [SetPrivateCreator\(\)](#).

12.314.3.10 IsPrivateCreator()

```
bool gdcm::Tag::IsPrivateCreator () const    [inline]
```

Returns if tag is a Private Creator (xxxx,00yy), where xxxx is odd number and yy in [0x10,0xFF]

Examples

[DuplicatePCDE.cxx](#).

References [GetElement\(\)](#), and [IsPrivate\(\)](#).

Referenced by [GetPrivateCreator\(\)](#).

12.314.3.11 IsPublic()

```
bool gdcm::Tag::IsPublic () const [inline]
```

STANDARD DATA ELEMENT: A Data [Element](#) defined in the DICOM Standard, and therefore listed in the DICOM Data [Element](#) Dictionary in PS 3.6. Is the [Tag](#) from the Public dict...well the implementation is buggy it does not prove the element is indeed in the dict...

Referenced by [IsPrivate\(\)](#).

12.314.3.12 operator"!="()

```
bool gdcm::Tag::operator!= (
    const Tag & _val) const [inline]
```

References [Tag\(\)](#), and [tag](#).

12.314.3.13 operator<()

```
bool gdcm::Tag::operator< (
    const Tag & _val) const [inline]
```

DICOM Standard expects the Data [Element](#) to be sorted by Tags All other comparison can be constructed from this one and operator ==

References [Tag\(\)](#), [tag](#), and [tags](#).

12.314.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
    const Tag & t2) const [inline]
```

References [Tag\(\)](#).

12.314.3.15 operator=()

```
Tag & gdcm::Tag::operator= (
    const Tag & _val) [inline]
```

References [Tag\(\)](#), and [tag](#).

12.314.3.16 operator==()

```
bool gdcm::Tag::operator== (
    const Tag & _val) const [inline]
```

References [Tag\(\)](#), and [tag](#).

12.314.3.17 operator[]() [1/2]

```
uint16_t & gdcM::Tag::operator[] (
    const unsigned int & _id) [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).

References [gdcM_assert](#).

12.314.3.18 operator[]() [2/2]

```
const uint16_t & gdcM::Tag::operator[] (
    const unsigned int & _id) const [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1).

References [gdcM_assert](#).

12.314.3.19 PrintAsContinuousString()

```
std::string gdcM::Tag::PrintAsContinuousString () const
```

Print tag value with no separating comma: eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

12.314.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcM::Tag::PrintAsContinuousUpperCaseString () const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

12.314.3.21 PrintAsPipeSeparatedString()

```
std::string gdcM::Tag::PrintAsPipeSeparatedString () const
```

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromPipeSeparatedString](#)

12.314.3.22 Read()

```
template<typename TSwap>
std::istream & gdcM::Tag::Read (
    std::istream & is) [inline]
```

Read a tag from binary representation.

12.314.3.23 ReadFromCommaSeparatedString()

```
bool gdcmm::Tag::ReadFromCommaSeparatedString (  
    const char * str)
```

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

12.314.3.24 ReadFromContinuousString()

```
bool gdcmm::Tag::ReadFromContinuousString (  
    const char * str)
```

Read From XML formatted tag value eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

12.314.3.25 ReadFromPipeSeparatedString()

```
bool gdcmm::Tag::ReadFromPipeSeparatedString (  
    const char * str)
```

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromCommaSeparatedString](#)

12.314.3.26 SetElement()

```
void gdcmm::Tag::SetElement (  
    uint16_t element) [inline]
```

Sets the 'Element number' of the given Tag.

Examples

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [gdcmm::PrivateTag::PrivateTag\(\)](#), [gdcmm::PrivateTag::PrivateTag\(\)](#), [GetPrivateCreator\(\)](#), [operator>>](#), and [SetPrivateCreator\(\)](#).

12.314.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (  
    uint16_t group,  
    uint16_t element) [inline]
```

Sets the 'Group number' & 'Element number' of the given [Tag](#).

Referenced by [Tag\(\)](#), and [gdcm::PrivateTag::operator=\(\)](#).

12.314.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (  
    uint32_t tag) [inline]
```

Sets the full tag value of the given [Tag](#).

References [tag](#).

12.314.3.29 SetGroup()

```
void gdcm::Tag::SetGroup (  
    uint16_t group) [inline]
```

Sets the 'Group number' of the given [Tag](#).

Referenced by [operator>>](#), and [SetPrivateCreator\(\)](#).

12.314.3.30 SetPrivateCreator()

```
void gdcm::Tag::SetPrivateCreator (  
    Tag const & t) [inline]
```

Set private creator:

Examples

[DuplicatePCDE.cxx](#).

References [Tag\(\)](#), [gdcm_assert](#), [GetElement\(\)](#), [GetGroup\(\)](#), [IsPrivate\(\)](#), [SetElement\(\)](#), and [SetGroup\(\)](#).

12.314.3.31 Write()

```
template<typename TSwap>  
const std::ostream & gdcm::Tag::Write (  
    std::ostream & os) const [inline]
```

Write a tag in binary rep.

Referenced by [gdcm::Item::Write\(\)](#), [gdcm::SequenceOfFragments::Write\(\)](#), and [gdcm::SequenceOfItems::Write\(\)](#).

12.314.4 Friends And Related Symbol Documentation

12.314.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const Tag & __val) [friend]
```

References [Tag\(\)](#).

12.314.4.2 operator>>

```
std::istream & operator>> (  
    std::istream & __is,  
    Tag & __val) [friend]
```

References [Tag\(\)](#), [SetElement\(\)](#), and [SetGroup\(\)](#).

12.314.5 Member Data Documentation

12.314.5.1 bytes

```
char gdcm::Tag::bytes[4]
```

12.314.5.2 tag

```
uint32_t gdcm::Tag::tag
```

Referenced by [Tag\(\)](#), [Tag\(\)](#), [operator!=\(\)](#), [operator<\(\)](#), [operator=\(\)](#), [operator==\(\(\)\)](#), and [SetElementTag\(\)](#).

12.314.5.3 tags

```
uint16_t gdcm::Tag::tags[2]
```

Referenced by [operator<\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmTag.h](#)

12.315 gdcm::TagPath Class Reference

class to handle a path of tag.

```
#include <gdcmTagPath.h>
```

Public Member Functions

- [TagPath](#) ()
- [~TagPath](#) ()
- bool [ConstructFromString](#) (const char *path)
- bool [ConstructFromTagList](#) ([Tag](#) const *l, unsigned int n)
Construct from a list of tags.
- void [Print](#) (std::ostream &) const
- bool [Push](#) ([Tag](#) const &t)
- bool [Push](#) (unsigned int itemnum)

Static Public Member Functions

- static bool [IsValid](#) (const char *path)
Return if path is valid or not.

12.315.1 Detailed Description

class to handle a path of tag.

Any Resemblance to Existing XPath is Purely Coincidental ftp://medical.nema.org/medical/dicom/supps/sup118_pc.pdf

12.315.2 Constructor & Destructor Documentation

12.315.2.1 TagPath()

gdcm::TagPath::TagPath ()

12.315.2.2 ~TagPath()

gdcm::TagPath::~~TagPath ()

12.315.3 Member Function Documentation

12.315.3.1 ConstructFromString()

bool gdcm::TagPath::ConstructFromString (
const char * path)

"/0018,0018/"... No space allowed, comma is use to separate tag group from tag element and slash is used to separate tag return false if invalid

12.315.3.2 ConstructFromTagList()

```
bool gdcm::TagPath::ConstructFromTagList (
    Tag const * l,
    unsigned int n)
```

Construct from a list of tags.

12.315.3.3 IsValid()

```
bool gdcm::TagPath::IsValid (
    const char * path) [static]
```

Return if path is valid or not.

12.315.3.4 Print()

```
void gdcm::TagPath::Print (
    std::ostream & ) const
```

12.315.3.5 Push() [1/2]

```
bool gdcm::TagPath::Push (
    Tag const & t)
```

12.315.3.6 Push() [2/2]

```
bool gdcm::TagPath::Push (
    unsigned int itemnum)
```

The documentation for this class was generated from the following file:

- [gdcmTagPath.h](#)

12.316 gdcm::Testing Class Reference

class for testing

```
#include <gdcmTesting.h>
```

Public Types

- typedef const char *const (* MD5DataImagesType)[2]
- typedef const char *const (* MediaStorageDataFilesType)[2]
return the table that map the media storage (as string) of a filename (gdcmData)

Public Member Functions

- [Testing](#) ()=default
- [~Testing](#) ()=default
- void [Print](#) (std::ostream &os=std::cout)
Print.

Static Public Member Functions

- static bool [ComputeFileMD5](#) (const char *filename, char digest_str[33])
- static bool [ComputeMD5](#) (const char *buffer, size_t buf_len, char digest_str[33])
- static const char * [GetDataExtraRoot](#) ()
Return the GDCM DATA EXTRA ROOT.
- static const char * [GetDataRoot](#) ()
Return the GDCM DATA ROOT.
- static const char * [GetFileName](#) (unsigned int file)
- static const char *const * [GetFileNames](#) ()
return the table of fullpath to gdcmData DICOM files:
- static int [GetLossyFlagFromFile](#) (const char *filepath)
- static const char *const * [GetMD5DataImage](#) (unsigned int file)
- static [MD5DataImagesType](#) [GetMD5DataImages](#) ()
- static const char * [GetMD5FromBrokenFile](#) (const char *filepath)
- static const char * [GetMD5FromFile](#) (const char *filepath)
- static const char *const * [GetMediaStorageDataFile](#) (unsigned int file)
- static [MediaStorageDataFilesType](#) [GetMediaStorageDataFiles](#) ()
- static const char * [GetMediaStorageFromFile](#) (const char *filepath)
- static unsigned int [GetNumberOfFileNames](#) ()
- static unsigned int [GetNumberOfMD5DataImages](#) ()
- static unsigned int [GetNumberOfMediaStorageDataFiles](#) ()
- static const char * [GetPixelSpacingDataRoot](#) ()
Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset).
- static std::streamoff [GetSelectedPrivateGroupOffsetFromFile](#) (const char *filepath)
- static std::streamoff [GetSelectedTagsOffsetFromFile](#) (const char *filepath)
- static const char * [GetSourceDirectory](#) ()
- static std::streamoff [GetStreamOffsetFromFile](#) (const char *filepath)
- static const char * [GetTempDirectory](#) (const char *subdir=nullptr)
- static const wchar_t * [GetTempDirectoryW](#) (const wchar_t *subdir=nullptr)
NOT THREAD SAFE.
- static const char * [GetTempFilename](#) (const char *filename, const char *subdir=nullptr)
NOT THREAD SAFE.
- static const wchar_t * [GetTempFilenameW](#) (const wchar_t *filename, const wchar_t *subdir=nullptr)
NOT THREAD SAFE.

12.316.1 Detailed Description

class for testing

this class is used for the nightly regression system for GDCM It makes heavily use of md5 computation

See also

[gdcm::MD5](#) class for md5 computation

12.316.2 Member Typedef Documentation

12.316.2.1 MD5DataImagesType

```
typedef const char* const(* gdcm::Testing::MD5DataImagesType)[2]
```

return the table that map the md5 (as in md5sum) of the Pixel Data associated to a filename

12.316.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

12.316.3 Constructor & Destructor Documentation

12.316.3.1 Testing()

```
gdcm::Testing::Testing () [default]
```

12.316.3.2 ~Testing()

```
gdcm::Testing::~~Testing () [default]
```

12.316.4 Member Function Documentation

12.316.4.1 ComputeFileMD5()

```
bool gdcm::Testing::ComputeFileMD5 (  
    const char * filename,  
    char digest_str[33]) [static]
```

Examples

[MetaImageMD5Activiz.cs](#).

12.316.4.2 ComputeMD5()

```
bool gdcm::Testing::ComputeMD5 (  
    const char * buffer,  
    size_t buf_len,  
    char digest_str[33]) [static]
```

[MD5](#) stuff digest_str needs to be at least : strlen = [2*16+1]; string will be \0 padded. (md5 are 32 bytes long) [Testing](#) is not meant to be shipped with an installed GDCM release, always prefer the [gdcm::MD5](#) API when doing md5 computation.

12.316.4.3 GetDataExtraRoot()

```
const char * gdcm::Testing::GetDataExtraRoot () [static]
```

Return the GDCM DATA EXTRA ROOT.

Examples

[DiscriminateVolume.cxx](#), [VolumeSorter.cxx](#), and [reslicesphere.cxx](#).

12.316.4.4 GetDataRoot()

```
const char * gdcm::Testing::GetDataRoot () [static]
```

Return the GDCM DATA ROOT.

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#),
and [MagnifyFile.cxx](#).

12.316.4.5 GetFileName()

```
const char * gdcm::Testing::GetFileName (
    unsigned int file) [static]
```

Examples

[MetaImageMD5Activiz.cs](#).

12.316.4.6 GetFileNames()

```
const char *const * gdcm::Testing::GetFileNames () [static]
```

return the table of fullpath to gdcmData DICOM files:

Examples

[TestReader.cxx](#).

12.316.4.7 GetLossyFlagFromFile()

```
int gdcm::Testing::GetLossyFlagFromFile (
    const char * filepath) [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

12.316.4.8 GetMD5DataImage()

```
const char *const * gdcmm::Testing::GetMD5DataImage (
    unsigned int file) [static]
```

12.316.4.9 GetMD5DataImages()

```
MD5DataImagesType gdcmm::Testing::GetMD5DataImages () [static]
```

12.316.4.10 GetMD5FromBrokenFile()

```
const char * gdcmm::Testing::GetMD5FromBrokenFile (
    const char * filepath) [static]
```

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

12.316.4.11 GetMD5FromFile()

```
const char * gdcmm::Testing::GetMD5FromFile (
    const char * filepath) [static]
```

12.316.4.12 GetMediaStorageDataFile()

```
const char *const * gdcmm::Testing::GetMediaStorageDataFile (
    unsigned int file) [static]
```

12.316.4.13 GetMediaStorageDataFiles()

```
MediaStorageDataFilesType gdcmm::Testing::GetMediaStorageDataFiles () [static]
```

12.316.4.14 GetMediaStorageFromFile()

```
const char * gdcmm::Testing::GetMediaStorageFromFile (
    const char * filepath) [static]
```

Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

12.316.4.15 GetNumberOfFileNames()

unsigned int gdcm::Testing::GetNumberOfFileNames () [static]

Examples

[MetaImageMD5Activiz.cs](#).

12.316.4.16 GetNumberOfMD5DataImages()

unsigned int gdcm::Testing::GetNumberOfMD5DataImages () [static]

12.316.4.17 GetNumberOfMediaStorageDataFiles()

unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles () [static]

12.316.4.18 GetPixelSpacingDataRoot()

const char * gdcm::Testing::GetPixelSpacingDataRoot () [static]

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset).

12.316.4.19 GetSelectedPrivateGroupOffsetFromFile()

std::streamoff gdcm::Testing::GetSelectedPrivateGroupOffsetFromFile (
const char * filepath) [static]

Return the offset just after private attribute (0009,0010,"GEMS_IDEN_01") if found. Otherwise the offset of the next attribute -1 if not found

12.316.4.20 GetSelectedTagsOffsetFromFile()

std::streamoff gdcm::Testing::GetSelectedTagsOffsetFromFile (
const char * filepath) [static]

Return the offset just after Pixel Data Length (7fe0,0000) if found. Otherwise the offset of the very first pixel cell in Pixel Data -1 if not found

12.316.4.21 GetSourceDirectory()

const char * gdcm::Testing::GetSourceDirectory () [static]

12.316.4.22 GetStreamOffsetFromFile()

```
std::streamoff gdcm::Testing::GetStreamOffsetFromFile (  
    const char * filepath) [static]
```

Return the offset of the very first pixel cell in the PixelData -1 if not found

12.316.4.23 GetTempDirectory()

```
const char * gdcm::Testing::GetTempDirectory (  
    const char * subdir = nullptr) [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples

[MetaImageMD5Activiz.cs](#).

12.316.4.24 GetTempDirectoryW()

```
const wchar_t * gdcm::Testing::GetTempDirectoryW (  
    const wchar_t * subdir = nullptr) [static]
```

NOT THREAD SAFE.

12.316.4.25 GetTempFilename()

```
const char * gdcm::Testing::GetTempFilename (  
    const char * filename,  
    const char * subdir = nullptr) [static]
```

NOT THREAD SAFE.

Examples

[MetaImageMD5Activiz.cs](#).

12.316.4.26 GetTempFilenameW()

```
const wchar_t * gdcm::Testing::GetTempFilenameW (  
    const wchar_t * filename,  
    const wchar_t * subdir = nullptr) [static]
```

NOT THREAD SAFE.

12.316.4.27 Print()

```
void gdcmm::Testing::Print (
    std::ostream & os = std::cout)
```

Print.

The documentation for this class was generated from the following file:

- [gdcmmTesting.h](#)

12.317 gdcmm::Trace Class Reference

[Trace](#).

```
#include <gdcmmTrace.h>
```

Public Member Functions

- [Trace](#) ()
- [~Trace](#) ()

Static Public Member Functions

- static void [DebugOff](#) ()
- static void [DebugOn](#) ()
- static void [ErrorOff](#) ()
- static void [ErrorOn](#) ()
- static bool [GetDebugFlag](#) ()
- static std::ostream & [GetDebugStream](#) ()
- static bool [GetErrorFlag](#) ()
- static std::ostream & [GetErrorStream](#) ()
- static std::ostream & [GetStream](#) ()
- static bool [GetWarningFlag](#) ()
- static std::ostream & [GetWarningStream](#) ()
- static void [SetDebug](#) (bool debug)

Turn debug messages on (default: false).
- static void [SetDebugStream](#) (std::ostream &os)

Explicitly set the stream which receive Debug messages:
- static void [SetError](#) (bool debug)

Turn error messages on (default: true).
- static void [SetErrorStream](#) (std::ostream &os)

Explicitly set the stream which receive Error messages:
- static void [SetStream](#) (std::ostream &os)
- static void [SetStreamToFile](#) (const char *filename)
- static void [SetWarning](#) (bool debug)

Turn warning messages on (default: true).
- static void [SetWarningStream](#) (std::ostream &os)

Explicitly set the stream which receive Warning messages:
- static void [WarningOff](#) ()
- static void [WarningOn](#) ()

12.317.1 Detailed Description

[Trace](#).

Debug / Warning and Error are encapsulated in this class by default the [Trace](#) class will redirect any debug/↔ warning/error to std::cerr. Unless SetStream was specified with another (open) stream or SetStreamToFile was specified to a writable file on the system.

Warning

All string messages are removed during compilation time when compiled with CMAKE_BUILD_TYPE being set to either:

- Release
- MinSizeRel It is recommended to compile with RelWithDebInfo and/or Debug during prototyping of applications.

Examples

[DecompressJPEGFile.cs](#).

12.317.2 Constructor & Destructor Documentation

12.317.2.1 Trace()

```
gdcmm::Trace::Trace ()
```

12.317.2.2 ~Trace()

```
gdcmm::Trace::~~Trace ()
```

12.317.3 Member Function Documentation

12.317.3.1 DebugOff()

```
void gdcmm::Trace::DebugOff () [static]
```

Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

12.317.3.2 DebugOn()

void gdcmm::Trace::DebugOn () [static]

Examples

[CreateFakePET.cxx](#), [DecompressJPEGFile.cs](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

12.317.3.3 ErrorOff()

void gdcmm::Trace::ErrorOff () [static]

Examples

[MetaImageMD5Activiz.cs](#).

12.317.3.4 ErrorOn()

void gdcmm::Trace::ErrorOn () [static]

12.317.3.5 GetDebugFlag()

bool gdcmm::Trace::GetDebugFlag () [static]

12.317.3.6 GetDebugStream()

std::ostream & gdcmm::Trace::GetDebugStream () [static]

12.317.3.7 GetErrorFlag()

bool gdcmm::Trace::GetErrorFlag () [static]

12.317.3.8 GetErrorStream()

std::ostream & gdcmm::Trace::GetErrorStream () [static]

12.317.3.9 GetStream()

std::ostream & gdcmm::Trace::GetStream () [static]

12.317.3.10 GetWarningFlag()

```
bool gdcm::Trace::GetWarningFlag () [static]
```

12.317.3.11 GetWarningStream()

```
std::ostream & gdcm::Trace::GetWarningStream () [static]
```

12.317.3.12 SetDebug()

```
void gdcm::Trace::SetDebug (  
    bool debug) [static]
```

Turn debug messages on (default: false).

Examples

[DumpToSQLITE3.cxx](#).

12.317.3.13 SetDebugStream()

```
void gdcm::Trace::SetDebugStream (  
    std::ostream & os) [static]
```

Explicitly set the stream which receive Debug messages:

12.317.3.14 SetError()

```
void gdcm::Trace::SetError (  
    bool debug) [static]
```

Turn error messages on (default: true).

12.317.3.15 SetErrorStream()

```
void gdcm::Trace::SetErrorStream (  
    std::ostream & os) [static]
```

Explicitly set the stream which receive Error messages:

Examples

[CStoreQtProgress.cxx](#).

12.317.3.16 SetStream()

```
void gdcm::Trace::SetStream (
    std::ostream & os) [static]
```

Explicitly set the ostream for [gdcm::Trace](#) to report to This will set the DebugStream, WarningStream and ErrorStream at once:

12.317.3.17 SetStreamToFile()

```
void gdcm::Trace::SetStreamToFile (
    const char * filename) [static]
```

Explicitly set the filename for [gdcm::Trace](#) to report to The file will be created (it will not append to existing file)

12.317.3.18 SetWarning()

```
void gdcm::Trace::SetWarning (
    bool debug) [static]
```

Turn warning messages on (default: true).

Examples

[DumpToSQLITE3.cxx](#).

12.317.3.19 SetWarningStream()

```
void gdcm::Trace::SetWarningStream (
    std::ostream & os) [static]
```

Explicitly set the stream which receive Warning messages:

12.317.3.20 WarningOff()

```
void gdcm::Trace::WarningOff () [static]
```

Examples

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

12.317.3.21 WarningOn()

```
void gdcm::Trace::WarningOn () [static]
```

Examples

[Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmTrace.h](#)

12.318 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

Public Types

- enum [NegociatedType](#) {
[Unknown](#) = 0 ,
[Explicit](#) ,
[Implicit](#) }
- enum [TSType](#) {
[ImplicitVRLittleEndian](#) = 0 ,
[ImplicitVRBigEndianPrivateGE](#) ,
[ExplicitVRLittleEndian](#) ,
[DeflatedExplicitVRLittleEndian](#) ,
[ExplicitVRBigEndian](#) ,
[JPEGBaselineProcess1](#) ,
[JPEGExtendedProcess2_4](#) ,
[JPEGExtendedProcess3_5](#) ,
[JPEGSpectralSelectionProcess6_8](#) ,
[JPEGFullProgressionProcess10_12](#) ,
[JPEGLosslessProcess14](#) ,
[JPEGLosslessProcess14_1](#) ,
[JPEGLSLossless](#) ,
[JPEGLSNearLossless](#) ,
[JPEG2000Lossless](#) ,
[JPEG2000](#) ,
[JPEG2000Part2Lossless](#) ,
[JPEG2000Part2](#) ,
[RLELossless](#) ,
[MPEG2MainProfile](#) ,
[ImplicitVRBigEndianACRNEMA](#) ,
[WeirdPapryus](#) ,
[CT_private_ELE](#) ,
[JPIPReferenced](#) ,

```

MPEG2MainProfileHighLevel ,
MPEG4AVCH264HighProfileLevel4_1 ,
MPEG4AVCH264BDcompatibleHighProfileLevel4_1 ,
HTJ2KLossless ,
HTJ2KRPCLLossless ,
HTJ2K ,
DeflatedImageFrameCompression ,
TS_END }

```

Public Member Functions

- [TransferSyntax](#) (TType type=[ImplicitVRLittleEndian](#))
- bool [CanStoreLossy](#) () const
- [NegociatedType](#) [GetNegociatedType](#) () const
- const char * [GetString](#) () const
- [SwapCode](#) [GetSwapCode](#) () const
- bool [IsEncapsulated](#) () const
- bool [IsEncoded](#) () const
- bool [IsExplicit](#) () const
- bool [IsImplicit](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsValid](#) () const
- operator TType () const

Static Public Member Functions

- static const char * [GetTSString](#) (TType ts)
- static TType [GetTSType](#) (const char *str)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [TransferSyntax](#) &ts)

12.318.1 Detailed Description

Class to manipulate Transfer Syntax.

Note

TRANSFER SYNTAX (Standard and Private): A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data [Element](#) structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

Todo : The implementation is completely retarded -> see [gdcn::UIDs](#) for a replacement We need: Is↔Supported We need preprocess of raw/xml file We need GetFullName()

Need a notion of Private Syntax. As defined in PS 3.5. Section 9.2

See also

[UIDs](#)

Examples

[BasicImageAnonymizer.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExplicitLittleEndian.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MpegVideoInfo.cs](#), and [StandardizeFiles.cs](#).

12.318.2 Member Enumeration Documentation

12.318.2.1 NegotiatedType

enum [gdcm::TransferSyntax::NegociatedType](#)

Enumerator

Unknown	
Explicit	
Implicit	

12.318.2.2 TSType

enum [gdcm::TransferSyntax::TSType](#)

Enumerator

ImplicitVRLittleEndian	
ImplicitVRBigEndianPrivateGE	
ExplicitVRLittleEndian	
DeflatedExplicitVRLittleEndian	
ExplicitVRBigEndian	
JPEGBaselineProcess1	
JPEGExtendedProcess2_4	
JPEGExtendedProcess3_5	
JPEGSpectralSelectionProcess6_8	
JPEGFullProgressionProcess10_12	
JPEGLosslessProcess14	

JPEGLosslessProcess14_1	
JPEGLSLossless	
JPEGLSNearLossless	
JPEG2000Lossless	
JPEG2000	
JPEG2000Part2Lossless	
JPEG2000Part2	
RLELossless	
MPEG2MainProfile	
ImplicitVRBigEndianACRNEMA	
WeirdPapryus	
CT_private_ELE	
JPIPReferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4_1	
HTJ2KLossless	
HTJ2KRPCLLossless	
HTJ2K	
DeflatedImageFrameCompression	
TS_END	

Examples

[BasicImageAnonymizer.cs](#), [CompressLossyJPEG.cs](#), [DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [ExplicitLittleEndian.cs](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [MpegVideoInfo.cs](#), and [StandardizeFiles.cs](#).

12.318.3 Constructor & Destructor Documentation

12.318.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
    TType type = ImplicitVRLittleEndian) [inline]
```

References [ImplicitVRLittleEndian](#).

Referenced by [operator<<](#).

12.318.4 Member Function Documentation

12.318.4.1 CanStoreLossy()

`bool gdcm::TransferSyntax::CanStoreLossy () const`

return true if TransFer Syntax Allow storing of Lossy Pixel Data

12.318.4.2 GetNegociatedType()

[NegociatedType](#) `gdcm::TransferSyntax::GetNegociatedType () const`

12.318.4.3 GetString()

`const char * gdcm::TransferSyntax::GetString () const` [inline]

References [GetTSString\(\)](#).

12.318.4.4 GetSwapCode()

[SwapCode](#) `gdcm::TransferSyntax::GetSwapCode () const`

Deprecated Return the [SwapCode](#) associated with the Transfer Syntax. Be careful with the special GE private syntax the [DataSet](#) is written in little endian but the Pixel Data is in Big Endian.

12.318.4.5 GetTSString()

`const char * gdcm::TransferSyntax::GetTSString (
 TSType ts) [static]`

Examples

[LargeVRDSExplicit.cxx](#).

Referenced by [GetString\(\)](#), and [operator<<](#).

12.318.4.6 GetTSType()

[TSType](#) `gdcm::TransferSyntax::GetTSType (
 const char * str) [static]`

12.318.4.7 IsEncapsulated()

bool gdcm::TransferSyntax::IsEncapsulated () const

Examples

[ExtractIconFromFile.cxx](#).

12.318.4.8 IsEncoded()

bool gdcm::TransferSyntax::IsEncoded () const

12.318.4.9 IsExplicit()

bool gdcm::TransferSyntax::IsExplicit () const

12.318.4.10 IsImplicit()

bool gdcm::TransferSyntax::IsImplicit () const

12.318.4.11 IsLossless()

bool gdcm::TransferSyntax::IsLossless () const

Return true if the transfer syntax algorithm is a lossless algorithm

12.318.4.12 IsLossy()

bool gdcm::TransferSyntax::IsLossy () const

Return true if the transfer syntax algorithm is a lossy algorithm

12.318.4.13 IsValid()

bool gdcm::TransferSyntax::IsValid () const [inline]

References [TS_END](#).

12.318.4.14 operator TSType()

gdcm::TransferSyntax::operator [TSType](#) () const [inline]

12.318.5 Friends And Related Symbol Documentation

12.318.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const TransferSyntax & ts) [friend]
```

References [TransferSyntax\(\)](#), and [GetTSString\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmTransferSyntax.h](#)

12.319 gdcm::network::TransferSyntaxSub Class Reference

[TransferSyntaxSub](#).

```
#include <gdcmTransferSyntaxSub.h>
```

Public Member Functions

- [TransferSyntaxSub](#) ()
- const char * [GetName](#) () const
- bool [operator==](#) (const [TransferSyntaxSub](#) &ts) const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetName](#) (const char *name)
- void [SetNameFromUID](#) ([UIDs::TSName](#) tsname)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.319.1 Detailed Description

[TransferSyntaxSub](#).

[Table 9-15](#) TRANSFER SYNTAX SUB-ITEM FIELDS

TODO what is the goal of :

[Table 9-19](#) TRANSFER SYNTAX SUB-ITEM FIELDS

12.319.2 Constructor & Destructor Documentation

12.319.2.1 TransferSyntaxSub()

`gdcm::network::TransferSyntaxSub::TransferSyntaxSub ()`

Referenced by [operator==\(\(\)\)](#).

12.319.3 Member Function Documentation

12.319.3.1 GetName()

`const char * gdcm::network::TransferSyntaxSub::GetName () const` [inline]

12.319.3.2 operator==(())

`bool gdcm::network::TransferSyntaxSub::operator==((const TransferSyntaxSub & ts) const` [inline]

References [TransferSyntaxSub\(\)](#).

12.319.3.3 Print()

`void gdcm::network::TransferSyntaxSub::Print (std::ostream & os) const`

12.319.3.4 Read()

`std::istream & gdcm::network::TransferSyntaxSub::Read (std::istream & is)`

12.319.3.5 SetName()

`void gdcm::network::TransferSyntaxSub::SetName (const char * name)`

12.319.3.6 SetNameFromUID()

`void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName tsname)`

12.319.3.7 Size()

```
size_t gdcm::network::TransferSyntaxSub::Size () const
```

12.319.3.8 Write()

```
const std::ostream & gdcm::network::TransferSyntaxSub::Write (  
    std::ostream & os) const
```

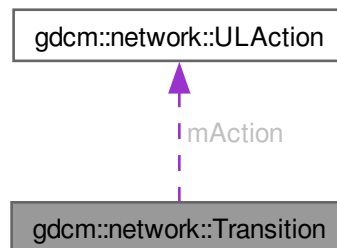
The documentation for this class was generated from the following file:

- [gdcmTransferSyntaxSub.h](#)

12.320 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::Transition:



Public Member Functions

- [Transition](#) ()
- [Transition](#) (int inEndState, [ULAction](#) *inAction)
- [~Transition](#) ()

Static Public Member Functions

- static [Transition](#) * [MakeNew](#) (int inEndState, [ULAction](#) *inAction)

Public Attributes

- [ULAction](#) * [mAction](#)
- int [mEnd](#)

12.320.1 Constructor & Destructor Documentation

12.320.1.1 Transition() [1/2]

`gdcmm::network::Transition::Transition ()` [inline]

References [gdcmm::network::eStaDoesNotExist](#), [mAction](#), and [mEnd](#).

Referenced by [MakeNew\(\)](#).

12.320.1.2 ~Transition()

`gdcmm::network::Transition::~~Transition ()` [inline]

References [mAction](#).

12.320.1.3 Transition() [2/2]

`gdcmm::network::Transition::Transition (
 int inEndState,
 ULAction * inAction)` [inline]

References [mAction](#), and [mEnd](#).

12.320.2 Member Function Documentation

12.320.2.1 MakeNew()

`Transition * gdcmm::network::Transition::MakeNew (
 int inEndState,
 ULAction * inAction)` [inline], [static]

References [Transition\(\)](#).

12.320.3 Member Data Documentation

12.320.3.1 mAction

`ULAction* gdcmm::network::Transition::mAction`

Referenced by [Transition\(\)](#), [Transition\(\)](#), and [~Transition\(\)](#).

12.320.3.2 mEnd

```
int gdcm::network::Transition::mEnd
```

Referenced by [Transition\(\)](#), and [Transition\(\)](#).

The documentation for this struct was generated from the following file:

- [gdcmULTransitionTable.h](#)

12.321 gdcm::Type Class Reference

[Type](#).

```
#include <gdcmType.h>
```

Public Types

- enum [TypeType](#) {
 [T1](#) = 0 ,
 [T1C](#) ,
 [T2](#) ,
 [T2C](#) ,
 [T3](#) ,
 [UNKNOWN](#) }

Public Member Functions

- [Type](#) ([TypeType](#) type=[UNKNOWN](#))
- [operator TypeType](#) () const

Static Public Member Functions

- static const char * [GetTypeString](#) ([TypeType](#) type)
- static [TypeType](#) [GetTypeType](#) (const char *type)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Type](#) &vr)

12.321.1 Detailed Description

[Type](#).

Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C
CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C
CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of [Type](#) 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

Examples

[TraverseModules.cxx](#).

12.321.2 Member Enumeration Documentation

12.321.2.1 TypeType

enum [gdcmm::Type::TypeType](#)

Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

12.321.3 Constructor & Destructor Documentation

12.321.3.1 Type()

[gdcmm::Type::Type](#) (
 [TypeType](#) type = [UNKNOWN](#)) [inline]

References [UNKNOWN](#).

Referenced by [operator<<](#).

12.321.4 Member Function Documentation

12.321.4.1 GetTypeString()

```
const char * gdcm::Type::GetTypeString (  
    TypeType type) [static]
```

Referenced by [operator<<](#).

12.321.4.2 GetTypeType()

```
TypeType gdcm::Type::GetTypeType (  
    const char * type) [static]
```

Referenced by [gdcm::ModuleEntry::ModuleEntry\(\)](#).

12.321.4.3 operator TypeType()

```
gdcm::Type::operator TypeType () const [inline]
```

12.321.5 Friends And Related Symbol Documentation

12.321.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Type & vr) [friend]
```

References [Type\(\)](#), and [GetTypeString\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmType.h](#)

12.322 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char [Internal](#) [64+1]

Friends

- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`

12.322.1 Friends And Related Symbol Documentation

12.322.1.1 `operator<<`

```
std::ostream & operator<< (
    std::ostream & _os,
    const UI & _val)    [friend]
```

References [Internal](#).

12.322.2 Member Data Documentation

12.322.2.1 Internal

```
char gdcmm::UI::Internal[64+1]
```

Referenced by [operator<<](#).

The documentation for this struct was generated from the following file:

- [gdcmmVR.h](#)

12.323 gdcmm::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcmmUIDGenerator.h>
```

Public Member Functions

- [UIDGenerator](#) ()
By default the root of a UID is a GDCM Root...
- const char * [Generate](#) ()

Static Public Member Functions

- static const char * [GetGDCMUID](#) ()
Return the default (GDCM) root UID:
- static const char * [GetRoot](#) ()
- static bool [IsValid](#) (const char *uid)
- static void [SetRoot](#) (const char *root)

Static Protected Member Functions

- static bool [GenerateUUID](#) (unsigned char *uuid_data)

12.323.1 Detailed Description

Class for generating unique UID.

When constructing a [Series](#) or [Study](#) UID, user has to keep around the UID, otherwise the UID Generator will simply forget the value and create a new UID.

Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [ManipulateFile.cs](#), [MpegVideoInfo.cs](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid_unique.cxx](#).

12.323.2 Constructor & Destructor Documentation

12.323.2.1 UIDGenerator()

```
gdcmm::UIDGenerator::UIDGenerator () [inline]
```

By default the root of a UID is a GDCM Root...

12.323.3 Member Function Documentation

12.323.3.1 Generate()

```
const char * gdcmm::UIDGenerator::Generate ()
```

Internally uses a std::string, so two calls have the same pointer ! save into a std::string In summary do not write code like that: const char *uid1 = uid.Generate(); const char *uid2 = uid.Generate(); since uid1 == uid2

Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [ManipulateFile.cs](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid_unique.cxx](#).

12.323.3.2 GenerateUUID()

```
bool gdcmm::UIDGenerator::GenerateUUID (
    unsigned char * uuid_data) [static], [protected]
```

12.323.3.3 GetGDCMUID()

```
const char * gdcm::UIDGenerator::GetGDCMUID () [static]
```

Return the default (GDCM) root UID:

12.323.3.4 GetRoot()

```
const char * gdcm::UIDGenerator::GetRoot () [static]
```

Examples

[ExplicitLittleEndian.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

12.323.3.5 IsValid()

```
bool gdcm::UIDGenerator::IsValid (  
    const char * uid) [static]
```

Find out if the string is a valid UID or not

[Todo](#) : Move that in `DataStructureAndEncoding` (see `FileMetaInformation::CheckFileMetaInformation`)

12.323.3.6 SetRoot()

```
void gdcm::UIDGenerator::SetRoot (  
    const char * root) [static]
```

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the [Generate\(\)](#) function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsablity for guaranteeing uniqueness of [UIDs](#) when the root is longer than 26 bytes.

Examples

[ExplicitLittleEndian.cs](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), and [uid_unique.cxx](#).

The documentation for this class was generated from the following file:

- [gdcmUIDGenerator.h](#)

12.324 gdcm::UIDs Class Reference

all known uids

```
#include <gdcmUIDs.h>
```

Public Types

- typedef const char *const ([TransferSyntaxStringsType](#))[2]
- enum [TSName](#) {
 - [VerificationSOPClass](#) = 1 ,
 - [ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#) = 2 ,
 - [ExplicitVRLittleEndian](#) = 3 ,
 - [DeflatedExplicitVRLittleEndian](#) = 4 ,
 - [ExplicitVRBigEndian](#) = 5 ,
 - [JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression](#) = 6 ,
 - [JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only](#) = 7 ,
 - [JPEGExtendedProcess35Retired](#) = 8 ,
 - [JPEGsSpectralSelectionNonHierarchicalProcess68Retired](#) = 9 ,
 - [JPEGsSpectralSelectionNonHierarchicalProcess79Retired](#) = 10 ,
 - [JPEGFullProgressionNonHierarchicalProcess1012Retired](#) = 11 ,
 - [JPEGFullProgressionNonHierarchicalProcess1113Retired](#) = 12 ,
 - [JPEGLosslessNonHierarchicalProcess14](#) = 13 ,
 - [JPEGLosslessNonHierarchicalProcess15Retired](#) = 14 ,
 - [JPEGExtendedHierarchicalProcess1618Retired](#) = 15 ,
 - [JPEGExtendedHierarchicalProcess1719Retired](#) = 16 ,
 - [JPEGsSpectralSelectionHierarchicalProcess2022Retired](#) = 17 ,
 - [JPEGsSpectralSelectionHierarchicalProcess2123Retired](#) = 18 ,
 - [JPEGFullProgressionHierarchicalProcess2426Retired](#) = 19 ,
 - [JPEGFullProgressionHierarchicalProcess2527Retired](#) = 20 ,
 - [JPEGLosslessHierarchicalProcess28Retired](#) = 21 ,
 - [JPEGLosslessHierarchicalProcess29Retired](#) = 22 ,
 - [JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression](#) = 23 ,
 - [JPEGLSLosslessImageCompression](#) = 24 ,
 - [JPEGLSLossyNearLosslessImageCompression](#) = 25 ,
 - [JPEG2000ImageCompressionLosslessOnly](#) = 26 ,
 - [JPEG2000ImageCompression](#) = 27 ,
 - [JPEG2000Part2MulticomponentImageCompressionLosslessOnly](#) = 28 ,
 - [JPEG2000Part2MulticomponentImageCompression](#) = 29 ,
 - [JPIPReferenced](#) = 30 ,
 - [JPIPReferencedDeflate](#) = 31 ,
 - [MPEG2MainProfileMainLevel](#) = 32 ,
 - [RLELossless](#) = 33 ,
 - [RFC2557MIMEencapsulation](#) = 34 ,
 - [XMLEncoding](#) = 35 ,
 - [MediaStorageDirectoryStorage](#) = 36 ,
 - [TalairachBrainAtlasFrameofReference](#) = 37 ,
 - [SPM2T1FrameofReference](#) = 38 ,
 - [SPM2T2FrameofReference](#) = 39 ,
 - [SPM2PDFFrameofReference](#) = 40 ,

[SPM2EPIFrameofReference](#) = 41 ,
[SPM2FILT1FrameofReference](#) = 42 ,
[SPM2PETFrameofReference](#) = 43 ,
[SPM2TRANSMFrameofReference](#) = 44 ,
[SPM2SPECTFrameofReference](#) = 45 ,
[SPM2GRAYFrameofReference](#) = 46 ,
[SPM2WHITEFrameofReference](#) = 47 ,
[SPM2CSFFrameofReference](#) = 48 ,
[SPM2BRAINMASKFrameofReference](#) = 49 ,
[SPM2AVG305T1FrameofReference](#) = 50 ,
[SPM2AVG152T1FrameofReference](#) = 51 ,
[SPM2AVG152T2FrameofReference](#) = 52 ,
[SPM2AVG152PDFFrameofReference](#) = 53 ,
[SPM2SINGLESUBJT1FrameofReference](#) = 54 ,
[ICBM452T1FrameofReference](#) = 55 ,
[ICBMSingleSubjectMRIFrameofReference](#) = 56 ,
[BasicStudyContentNotificationSOPClassRetired](#) = 57 ,
[StorageCommitmentPushModelSOPClass](#) = 58 ,
[StorageCommitmentPushModelSOPInstance](#) = 59 ,
[StorageCommitmentPullModelSOPClassRetired](#) = 60 ,
[StorageCommitmentPullModelSOPInstanceRetired](#) = 61 ,
[ProceduralEventLoggingSOPClass](#) = 62 ,
[ProceduralEventLoggingSOPInstance](#) = 63 ,
[SubstanceAdministrationLoggingSOPClass](#) = 64 ,
[SubstanceAdministrationLoggingSOPInstance](#) = 65 ,
[DICOMUIDRegistry](#) = 66 ,
[DICOMControlledTerminology](#) = 67 ,
[DICOMApplicationContextName](#) = 68 ,
[DetachedPatientManagementSOPClassRetired](#) = 69 ,
[DetachedPatientManagementMetaSOPClassRetired](#) = 70 ,
[DetachedVisitManagementSOPClassRetired](#) = 71 ,
[DetachedStudyManagementSOPClassRetired](#) = 72 ,
[StudyComponentManagementSOPClassRetired](#) = 73 ,
[ModalityPerformedProcedureStepSOPClass](#) = 74 ,
[ModalityPerformedProcedureStepRetrieveSOPClass](#) = 75 ,
[ModalityPerformedProcedureStepNotificationSOPClass](#) = 76 ,
[DetachedResultsManagementSOPClassRetired](#) = 77 ,
[DetachedResultsManagementMetaSOPClassRetired](#) = 78 ,
[DetachedStudyManagementMetaSOPClassRetired](#) = 79 ,
[DetachedInterpretationManagementSOPClassRetired](#) = 80 ,
[StorageServiceClass](#) = 81 ,
[BasicFilmSessionSOPClass](#) = 82 ,
[BasicFilmBoxSOPClass](#) = 83 ,
[BasicGrayscaleImageBoxSOPClass](#) = 84 ,
[BasicColorImageBoxSOPClass](#) = 85 ,
[ReferencedImageBoxSOPClassRetired](#) = 86 ,
[BasicGrayscalePrintManagementMetaSOPClass](#) = 87 ,
[ReferencedGrayscalePrintManagementMetaSOPClassRetired](#) = 88 ,
[PrintJobSOPClass](#) = 89 ,
[BasicAnnotationBoxSOPClass](#) = 90 ,
[PrinterSOPClass](#) = 91 ,
[PrinterConfigurationRetrievalSOPClass](#) = 92 ,
[PrinterSOPInstance](#) = 93 ,
[PrinterConfigurationRetrievalSOPInstance](#) = 94 ,

[BasicColorPrintManagementMetaSOPClass](#) = 95 ,
[ReferencedColorPrintManagementMetaSOPClassRetired](#) = 96 ,
[VOILUTBoxSOPClass](#) = 97 ,
[PresentationLUTSOPClass](#) = 98 ,
[ImageOverlayBoxSOPClassRetired](#) = 99 ,
[BasicPrintImageOverlayBoxSOPClassRetired](#) = 100 ,
[PrintQueueSOPInstanceRetired](#) = 101 ,
[PrintQueueManagementSOPClassRetired](#) = 102 ,
[StoredPrintStorageSOPClassRetired](#) = 103 ,
[HardcopyGrayscaleImageStorageSOPClassRetired](#) = 104 ,
[HardcopyColorImageStorageSOPClassRetired](#) = 105 ,
[PullPrintRequestSOPClassRetired](#) = 106 ,
[PullStoredPrintManagementMetaSOPClassRetired](#) = 107 ,
[MediaCreationManagementSOPClassUID](#) = 108 ,
[ComputedRadiographyImageStorage](#) = 109 ,
[DigitalXRayImageStorageForPresentation](#) = 110 ,
[DigitalXRayImageStorageForProcessing](#) = 111 ,
[DigitalMammographyXRayImageStorageForPresentation](#) = 112 ,
[DigitalMammographyXRayImageStorageForProcessing](#) = 113 ,
[DigitalIntraoralXRayImageStorageForPresentation](#) = 114 ,
[DigitalIntraoralXRayImageStorageForProcessing](#) = 115 ,
[CTImageStorage](#) = 116 ,
[EnhancedCTImageStorage](#) = 117 ,
[UltrasoundMultiframeImageStorageRetired](#) = 118 ,
[UltrasoundMultiframeImageStorage](#) = 119 ,
[MRImageStorage](#) = 120 ,
[EnhancedMRImageStorage](#) = 121 ,
[MRSpectroscopyStorage](#) = 122 ,
[NuclearMedicineImageStorageRetired](#) = 123 ,
[UltrasoundImageStorageRetired](#) = 124 ,
[UltrasoundImageStorage](#) = 125 ,
[SecondaryCaptureImageStorage](#) = 126 ,
[MultiframeSingleBitSecondaryCaptureImageStorage](#) = 127 ,
[MultiframeGrayscaleByteSecondaryCaptureImageStorage](#) = 128 ,
[MultiframeGrayscaleWordSecondaryCaptureImageStorage](#) = 129 ,
[MultiframeTrueColorSecondaryCaptureImageStorage](#) = 130 ,
[StandaloneOverlayStorageRetired](#) = 131 ,
[StandaloneCurveStorageRetired](#) = 132 ,
[WaveformStorageTrialRetired](#) = 133 ,
[ECG12leadWaveformStorage](#) = 134 ,
[GeneralECGWaveformStorage](#) = 135 ,
[AmbulatoryECGWaveformStorage](#) = 136 ,
[HemodynamicWaveformStorage](#) = 137 ,
[CardiacElectrophysiologyWaveformStorage](#) = 138 ,
[BasicVoiceAudioWaveformStorage](#) = 139 ,
[StandaloneModalityLUTStorageRetired](#) = 140 ,
[StandaloneVOILUTStorageRetired](#) = 141 ,
[GrayscaleSoftcopyPresentationStateStorageSOPClass](#) = 142 ,
[ColorSoftcopyPresentationStateStorageSOPClass](#) = 143 ,
[PseudoColorSoftcopyPresentationStateStorageSOPClass](#) = 144 ,
[BlendingSoftcopyPresentationStateStorageSOPClass](#) = 145 ,
[XRayAngiographicImageStorage](#) = 146 ,
[EnhancedXAImageStorage](#) = 147 ,
[XRayRadiofluoroscopicImageStorage](#) = 148 ,

EnhancedXRFImageStorage = 149 ,
XRay3DAngiographicImageStorage = 150 ,
XRay3DCraniofacialImageStorage = 151 ,
XRayAngiographicBiPlaneImageStorageRetired = 152 ,
NuclearMedicineImageStorage = 153 ,
RawDataStorage = 154 ,
SpatialRegistrationStorage = 155 ,
SpatialFiducialsStorage = 156 ,
DeformableSpatialRegistrationStorage = 157 ,
SegmentationStorage = 158 ,
RealWorldValueMappingStorage = 159 ,
VLImageStorageTrialRetired = 160 ,
VLMultiframeImageStorageTrialRetired = 161 ,
VLEndoscopicImageStorage = 162 ,
VideoEndoscopicImageStorage = 163 ,
VLMicroscopicImageStorage = 164 ,
VideoMicroscopicImageStorage = 165 ,
VLSlideCoordinatesMicroscopicImageStorage = 166 ,
VLPhotographicImageStorage = 167 ,
VideoPhotographicImageStorage = 168 ,
OphthalmicPhotography8BitImageStorage = 169 ,
OphthalmicPhotography16BitImageStorage = 170 ,
StereometricRelationshipStorage = 171 ,
OphthalmicTomographyImageStorage = 172 ,
TextSRStorageTrialRetired = 173 ,
AudioSRStorageTrialRetired = 174 ,
DetailSRStorageTrialRetired = 175 ,
ComprehensiveSRStorageTrialRetired = 176 ,
BasicTextSRStorage = 177 ,
EnhancedSRStorage = 178 ,
ComprehensiveSRStorage = 179 ,
ProcedureLogStorage = 180 ,
MammographyCADSRStorage = 181 ,
KeyObjectSelectionDocumentStorage = 182 ,
ChestCADSRStorage = 183 ,
XRayRadiationDoseSRStorage = 184 ,
EncapsulatedPDFStorage = 185 ,
EncapsulatedCDASStorage = 186 ,
PositronEmissionTomographyImageStorage = 187 ,
StandalonePETCurveStorageRetired = 188 ,
RTImageStorage = 189 ,
RTDoseStorage = 190 ,
RTStructureSetStorage = 191 ,
RTBeamsTreatmentRecordStorage = 192 ,
RTPlanStorage = 193 ,
RTBrachyTreatmentRecordStorage = 194 ,
RTTreatmentSummaryRecordStorage = 195 ,
RTIonPlanStorage = 196 ,
RTIonBeamsTreatmentRecordStorage = 197 ,
PatientRootQueryRetrieveInformationModelFIND = 198 ,
PatientRootQueryRetrieveInformationModelMOVE = 199 ,
PatientRootQueryRetrieveInformationModelGET = 200 ,
StudyRootQueryRetrieveInformationModelFIND = 201 ,
StudyRootQueryRetrieveInformationModelMOVE = 202 ,

[StudyRootQueryRetrieveInformationModelGET](#) = 203 ,
[PatientStudyOnlyQueryRetrieveInformationModelFINDRetired](#) = 204 ,
[PatientStudyOnlyQueryRetrieveInformationModelMOVERetired](#) = 205 ,
[PatientStudyOnlyQueryRetrieveInformationModelGETRetired](#) = 206 ,
[ModalityWorklistInformationModelFIND](#) = 207 ,
[GeneralPurposeWorklistInformationModelFIND](#) = 208 ,
[GeneralPurposeScheduledProcedureStepSOPClass](#) = 209 ,
[GeneralPurposePerformedProcedureStepSOPClass](#) = 210 ,
[GeneralPurposeWorklistManagementMetaSOPClass](#) = 211 ,
[InstanceAvailabilityNotificationSOPClass](#) = 212 ,
[RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft](#) = 213 ,
[RTConventionalMachineVerificationSupplement74FrozenDraft](#) = 214 ,
[RTIonMachineVerificationSupplement74FrozenDraft](#) = 215 ,
[UnifiedWorklistandProcedureStepServiceClass](#) = 216 ,
[UnifiedProcedureStepPushSOPClass](#) = 217 ,
[UnifiedProcedureStepWatchSOPClass](#) = 218 ,
[UnifiedProcedureStepPullSOPClass](#) = 219 ,
[UnifiedProcedureStepEventSOPClass](#) = 220 ,
[UnifiedWorklistandProcedureStepSOPInstance](#) = 221 ,
[GeneralRelevantPatientInformationQuery](#) = 222 ,
[BreastImagingRelevantPatientInformationQuery](#) = 223 ,
[CardiacRelevantPatientInformationQuery](#) = 224 ,
[HangingProtocolStorage](#) = 225 ,
[HangingProtocolInformationModelFIND](#) = 226 ,
[HangingProtocolInformationModelMOVE](#) = 227 ,
[ProductCharacteristicsQuerySOPClass](#) = 228 ,
[SubstanceApprovalQuerySOPClass](#) = 229 ,
[dicomDeviceName](#) = 230 ,
[dicomDescription](#) = 231 ,
[dicomManufacturer](#) = 232 ,
[dicomManufacturerModelName](#) = 233 ,
[dicomSoftwareVersion](#) = 234 ,
[dicomVendorData](#) = 235 ,
[dicomAETitle](#) = 236 ,
[dicomNetworkConnectionReference](#) = 237 ,
[dicomApplicationCluster](#) = 238 ,
[dicomAssociationInitiator](#) = 239 ,
[dicomAssociationAcceptor](#) = 240 ,
[dicomHostname](#) = 241 ,
[dicomPort](#) = 242 ,
[dicomSOPClass](#) = 243 ,
[dicomTransferRole](#) = 244 ,
[dicomTransferSyntax](#) = 245 ,
[dicomPrimaryDeviceType](#) = 246 ,
[dicomRelatedDeviceReference](#) = 247 ,
[dicomPreferredCalledAETitle](#) = 248 ,
[dicomTLSCyphersuite](#) = 249 ,
[dicomAuthorizedNodeCertificateReference](#) = 250 ,
[dicomThisNodeCertificateReference](#) = 251 ,
[dicomInstalled](#) = 252 ,
[dicomStationName](#) = 253 ,
[dicomDeviceSerialNumber](#) = 254 ,
[dicomInstitutionName](#) = 255 ,
[dicomInstitutionAddress](#) = 256 ,

[dicomInstitutionDepartmentName](#) = 257 ,
[dicomIssuerOfPatientID](#) = 258 ,
[dicomPreferredCallingAETitle](#) = 259 ,
[dicomSupportedCharacterSet](#) = 260 ,
[dicomConfigurationRoot](#) = 261 ,
[dicomDevicesRoot](#) = 262 ,
[dicomUniqueAETitlesRegistryRoot](#) = 263 ,
[dicomDevice](#) = 264 ,
[dicomNetworkAE](#) = 265 ,
[dicomNetworkConnection](#) = 266 ,
[dicomUniqueAETitle](#) = 267 ,
[dicomTransferCapability](#) = 268 ,
[VLWholeSlideMicroscopyImageStorage](#) = 269 ,
[EnhancedUSVolumeStorage](#) = 270 ,
[SurfaceSegmentationStorage](#) = 271 ,
[BreastTomosynthesisImageStorage](#) = 272 ,
[LegacyConvertedEnhancedCTImageStorage](#) = 273 ,
[LegacyConvertedEnhancedMRIImageStorage](#) = 274 ,
[LegacyConvertedEnhancedPETImageStorage](#) = 275 ,
[MPEG2MainProfileHighLevel](#) = 276 ,
[MPEG4AVCH_264HighProfileLevel4_1](#) = 277 ,
[MPEG4AVCH_264BDcompatibleHighProfileLevel4_1](#) = 278 ,
[PETColorPaletteSOPInstance](#) = 279 ,
[HotMetalBlueColorPaletteSOPInstance](#) = 280 ,
[PET20StepColorPaletteSOPInstance](#) = 281 ,
[SpringColorPaletteSOPInstance](#) = 282 ,
[SummerColorPaletteSOPInstance](#) = 283 ,
[FallColorPaletteSOPInstance](#) = 284 ,
[WinterColorPaletteSOPInstance](#) = 285 ,
[Papyrus3ImplicitVRLittleEndian](#) = 286 ,
[AdultMouseAnatomyOntology](#) = 287 ,
[UberonOntology](#) = 288 ,
[IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN](#) = 289 ,
[MouseGenomeInitiativeMGI](#) = 290 ,
[PubChemCompoundCID](#) = 291 ,
[ICD11](#) = 292 ,
[NewYorkUniversityMelanomaClinicalCooperativeGroup](#) = 293 ,
[MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide](#) = 294 ,
[ImageBiomarkerStandardisationInitiative](#) = 295 ,
[RadiomicsOntology](#) = 296 ,
[DisplaySystemSOPClass](#) = 297 ,
[DisplaySystemSOPInstance](#) = 298 ,
[GeneralAudioWaveformStorage](#) = 299 ,
[ArterialPulseWaveformStorage](#) = 300 ,
[RespiratoryWaveformStorage](#) = 301 ,
[XAXRFGayscaleSoftcopyPresentationStateStorage](#) = 302 ,
[GrayscalePlanarMPRVolumetricPresentationStateStorage](#) = 303 ,
[MPEG4AVCH_264HighProfileLevel4_2For2DVideo](#) = 304 ,
[MPEG4AVCH_264HighProfileLevel4_2For3DVideo](#) = 305 ,
[MPEG4AVCH_264StereoHighProfileLevel4_2](#) = 306 ,
[HEVCH_265MainProfileLevel5_1](#) = 307 ,
[HEVCH_265Main10ProfileLevel5_1](#) = 308 ,
[HotIronColorPaletteSOPInstance](#) = 309 ,
[CompositingPlanarMPRVolumetricPresentationStateStorage](#) = 310 ,

[AdvancedBlendingPresentationStateStorage](#) = 311 ,
[VolumeRenderingVolumetricPresentationStateStorage](#) = 312 ,
[SegmentedVolumeRenderingVolumetricPresentationStateStorage](#) = 313 ,
[MultipleVolumeRenderingVolumetricPresentationStateStorage](#) = 314 ,
[Null0](#) = 315 ,
[BreastProjectionXRayImageStorageForPresentation](#) = 316 ,
[BreastProjectionXRayImageStorageForProcessing](#) = 317 ,
[IntravascularOpticalCoherenceTomographyImageStorageForPresentation](#) = 318 ,
[IntravascularOpticalCoherenceTomographyImageStorageForProcessing](#) = 319 ,
[ParametricMapStorage](#) = 320 ,
[Null1](#) = 321 ,
[TractographyResultsStorage](#) = 322 ,
[SurfaceScanMeshStorage](#) = 323 ,
[SurfaceScanPointCloudStorage](#) = 324 ,
[WideFieldOphthalmicPhotographyStereographicProjectionImageStorage](#) = 325 ,
[WideFieldOphthalmicPhotography3DCoordinatesImageStorage](#) = 326 ,
[OphthalmicOpticalCoherenceTomographyEnFaceImageStorage](#) = 327 ,
[OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage](#) = 328 ,
[LensometryMeasurementsStorage](#) = 329 ,
[AutorefractionMeasurementsStorage](#) = 330 ,
[KeratometryMeasurementsStorage](#) = 331 ,
[SubjectiveRefractionMeasurementsStorage](#) = 332 ,
[VisualAcuityMeasurementsStorage](#) = 333 ,
[SpectaclePrescriptionReportStorage](#) = 334 ,
[OphthalmicAxialMeasurementsStorage](#) = 335 ,
[IntraocularLensCalculationsStorage](#) = 336 ,
[MacularGridThicknessandVolumeReportStorage](#) = 337 ,
[OphthalmicVisualFieldStaticPerimetryMeasurementsStorage](#) = 338 ,
[OphthalmicThicknessMapStorage](#) = 339 ,
[CornealTopographyMapStorage](#) = 340 ,
[Comprehensive3DSRStorage](#) = 341 ,
[ExtensibleSRStorage](#) = 342 ,
[RadiopharmaceuticalRadiationDoseSRStorage](#) = 343 ,
[ColonCADSRStorage](#) = 344 ,
[ImplantationPlanSRStorage](#) = 345 ,
[AcquisitionContextSRStorage](#) = 346 ,
[SimplifiedAdultEchoSRStorage](#) = 347 ,
[PatientRadiationDoseSRStorage](#) = 348 ,
[PlannedImagingAgentAdministrationSRStorage](#) = 349 ,
[PerformedImagingAgentAdministrationSRStorage](#) = 350 ,
[ContentAssessmentResultsStorage](#) = 351 ,
[EncapsulatedSTLStorage](#) = 352 ,
[EnhancedPETImageStorage](#) = 353 ,
[BasicStructuredDisplayStorage](#) = 354 ,
[CTDefinedProcedureProtocolStorage](#) = 355 ,
[CTPerformedProcedureProtocolStorage](#) = 356 ,
[ProtocolApprovalStorage](#) = 357 ,
[ProtocolApprovalInformationModelFIND](#) = 358 ,
[ProtocolApprovalInformationModelMOVE](#) = 359 ,
[ProtocolApprovalInformationModelGET](#) = 360 ,
[RTPhysicianIntentStorage](#) = 361 ,
[RTSegmentAnnotationStorage](#) = 362 ,
[DICOSCTImageStorage](#) = 363 ,
[DICOSDigitalXRayImageStorageForPresentation](#) = 364 ,

```

DICOSDigitalXRayImageStorageForProcessing = 365 ,
DICOSThreatDetectionReportStorage = 366 ,
DICOS2DAITStorage = 367 ,
DICOS3DAITStorage = 368 ,
DICOSQuadrupoleResonanceQRStorage = 369 ,
EddyCurrentImageStorage = 370 ,
EddyCurrentMultiframeImageStorage = 371 ,
CompositeInstanceRootRetrieveMOVE = 372 ,
CompositeInstanceRootRetrieveGET = 373 ,
CompositeInstanceRetrieveWithoutBulkDataGET = 374 ,
DefinedProcedureProtocolInformationModelFIND = 375 ,
DefinedProcedureProtocolInformationModelMOVE = 376 ,
DefinedProcedureProtocolInformationModelGET = 377 ,
UPSFilteredGlobalSubscriptionSOPInstance = 378 ,
UnifiedWorklistandProcedureStepServiceClass1 = 379 ,
UnifiedProcedureStepPushSOPClass1 = 380 ,
UnifiedProcedureStepWatchSOPClass1 = 381 ,
UnifiedProcedureStepPullSOPClass1 = 382 ,
UnifiedProcedureStepEventSOPClass1 = 383 ,
RTBeamsDeliveryInstructionStorage = 384 ,
RTConventionalMachineVerification = 385 ,
RTIonMachineVerification = 386 ,
RTBrachyApplicationSetupDeliveryInstructionStorage = 387 ,
HangingProtocolInformationModelGET = 388 ,
ColorPaletteStorage = 389 ,
ColorPaletteQueryRetrieveInformationModelFIND = 390 ,
ColorPaletteQueryRetrieveInformationModelMOVE = 391 ,
ColorPaletteQueryRetrieveInformationModelGET = 392 ,
GenericImplantTemplateStorage = 393 ,
GenericImplantTemplateInformationModelFIND = 394 ,
GenericImplantTemplateInformationModelMOVE = 395 ,
GenericImplantTemplateInformationModelGET = 396 ,
ImplantAssemblyTemplateStorage = 397 ,
ImplantAssemblyTemplateInformationModelFIND = 398 ,
ImplantAssemblyTemplateInformationModelMOVE = 399 ,
ImplantAssemblyTemplateInformationModelGET = 400 ,
ImplantTemplateGroupStorage = 401 ,
ImplantTemplateGroupInformationModelFIND = 402 ,
ImplantTemplateGroupInformationModelMOVE = 403 ,
ImplantTemplateGroupInformationModelGET = 404 ,
NativeDICOMModel = 405 ,
AbstractMultiDimensionalImageModel = 406 ,
DICOMContentMappingResource = 407 ,
EnhancedMRColorImageStorage = 408 ,
UniversalCoordinatedTime = 409 }
• enum TSType {
uid_1_2_840_10008_1_1 = 1 ,
uid_1_2_840_10008_1_2 = 2 ,
uid_1_2_840_10008_1_2_1 = 3 ,
uid_1_2_840_10008_1_2_1_99 = 4 ,
uid_1_2_840_10008_1_2_2 = 5 ,
uid_1_2_840_10008_1_2_4_50 = 6 ,
uid_1_2_840_10008_1_2_4_51 = 7 ,
uid_1_2_840_10008_1_2_4_52 = 8 ,

```

uid_1_2_840_10008_1_2_4_53 = 9 ,
uid_1_2_840_10008_1_2_4_54 = 10 ,
uid_1_2_840_10008_1_2_4_55 = 11 ,
uid_1_2_840_10008_1_2_4_56 = 12 ,
uid_1_2_840_10008_1_2_4_57 = 13 ,
uid_1_2_840_10008_1_2_4_58 = 14 ,
uid_1_2_840_10008_1_2_4_59 = 15 ,
uid_1_2_840_10008_1_2_4_60 = 16 ,
uid_1_2_840_10008_1_2_4_61 = 17 ,
uid_1_2_840_10008_1_2_4_62 = 18 ,
uid_1_2_840_10008_1_2_4_63 = 19 ,
uid_1_2_840_10008_1_2_4_64 = 20 ,
uid_1_2_840_10008_1_2_4_65 = 21 ,
uid_1_2_840_10008_1_2_4_66 = 22 ,
uid_1_2_840_10008_1_2_4_70 = 23 ,
uid_1_2_840_10008_1_2_4_80 = 24 ,
uid_1_2_840_10008_1_2_4_81 = 25 ,
uid_1_2_840_10008_1_2_4_90 = 26 ,
uid_1_2_840_10008_1_2_4_91 = 27 ,
uid_1_2_840_10008_1_2_4_92 = 28 ,
uid_1_2_840_10008_1_2_4_93 = 29 ,
uid_1_2_840_10008_1_2_4_94 = 30 ,
uid_1_2_840_10008_1_2_4_95 = 31 ,
uid_1_2_840_10008_1_2_4_100 = 32 ,
uid_1_2_840_10008_1_2_5 = 33 ,
uid_1_2_840_10008_1_2_6_1 = 34 ,
uid_1_2_840_10008_1_2_6_2 = 35 ,
uid_1_2_840_10008_1_3_10 = 36 ,
uid_1_2_840_10008_1_4_1_1 = 37 ,
uid_1_2_840_10008_1_4_1_2 = 38 ,
uid_1_2_840_10008_1_4_1_3 = 39 ,
uid_1_2_840_10008_1_4_1_4 = 40 ,
uid_1_2_840_10008_1_4_1_5 = 41 ,
uid_1_2_840_10008_1_4_1_6 = 42 ,
uid_1_2_840_10008_1_4_1_7 = 43 ,
uid_1_2_840_10008_1_4_1_8 = 44 ,
uid_1_2_840_10008_1_4_1_9 = 45 ,
uid_1_2_840_10008_1_4_1_10 = 46 ,
uid_1_2_840_10008_1_4_1_11 = 47 ,
uid_1_2_840_10008_1_4_1_12 = 48 ,
uid_1_2_840_10008_1_4_1_13 = 49 ,
uid_1_2_840_10008_1_4_1_14 = 50 ,
uid_1_2_840_10008_1_4_1_15 = 51 ,
uid_1_2_840_10008_1_4_1_16 = 52 ,
uid_1_2_840_10008_1_4_1_17 = 53 ,
uid_1_2_840_10008_1_4_1_18 = 54 ,
uid_1_2_840_10008_1_4_2_1 = 55 ,
uid_1_2_840_10008_1_4_2_2 = 56 ,
uid_1_2_840_10008_1_9 = 57 ,
uid_1_2_840_10008_1_20_1 = 58 ,
uid_1_2_840_10008_1_20_1_1 = 59 ,
uid_1_2_840_10008_1_20_2 = 60 ,
uid_1_2_840_10008_1_20_2_1 = 61 ,
uid_1_2_840_10008_1_40 = 62 ,

```
uid_1_2_840_10008_1_40_1 = 63 ,  
uid_1_2_840_10008_1_42 = 64 ,  
uid_1_2_840_10008_1_42_1 = 65 ,  
uid_1_2_840_10008_2_6_1 = 66 ,  
uid_1_2_840_10008_2_16_4 = 67 ,  
uid_1_2_840_10008_3_1_1_1 = 68 ,  
uid_1_2_840_10008_3_1_2_1_1 = 69 ,  
uid_1_2_840_10008_3_1_2_1_4 = 70 ,  
uid_1_2_840_10008_3_1_2_2_1 = 71 ,  
uid_1_2_840_10008_3_1_2_3_1 = 72 ,  
uid_1_2_840_10008_3_1_2_3_2 = 73 ,  
uid_1_2_840_10008_3_1_2_3_3 = 74 ,  
uid_1_2_840_10008_3_1_2_3_4 = 75 ,  
uid_1_2_840_10008_3_1_2_3_5 = 76 ,  
uid_1_2_840_10008_3_1_2_5_1 = 77 ,  
uid_1_2_840_10008_3_1_2_5_4 = 78 ,  
uid_1_2_840_10008_3_1_2_5_5 = 79 ,  
uid_1_2_840_10008_3_1_2_6_1 = 80 ,  
uid_1_2_840_10008_4_2 = 81 ,  
uid_1_2_840_10008_5_1_1_1 = 82 ,  
uid_1_2_840_10008_5_1_1_2 = 83 ,  
uid_1_2_840_10008_5_1_1_4 = 84 ,  
uid_1_2_840_10008_5_1_1_4_1 = 85 ,  
uid_1_2_840_10008_5_1_1_4_2 = 86 ,  
uid_1_2_840_10008_5_1_1_9 = 87 ,  
uid_1_2_840_10008_5_1_1_9_1 = 88 ,  
uid_1_2_840_10008_5_1_1_14 = 89 ,  
uid_1_2_840_10008_5_1_1_15 = 90 ,  
uid_1_2_840_10008_5_1_1_16 = 91 ,  
uid_1_2_840_10008_5_1_1_16_376 = 92 ,  
uid_1_2_840_10008_5_1_1_17 = 93 ,  
uid_1_2_840_10008_5_1_1_17_376 = 94 ,  
uid_1_2_840_10008_5_1_1_18 = 95 ,  
uid_1_2_840_10008_5_1_1_18_1 = 96 ,  
uid_1_2_840_10008_5_1_1_22 = 97 ,  
uid_1_2_840_10008_5_1_1_23 = 98 ,  
uid_1_2_840_10008_5_1_1_24 = 99 ,  
uid_1_2_840_10008_5_1_1_24_1 = 100 ,  
uid_1_2_840_10008_5_1_1_25 = 101 ,  
uid_1_2_840_10008_5_1_1_26 = 102 ,  
uid_1_2_840_10008_5_1_1_27 = 103 ,  
uid_1_2_840_10008_5_1_1_29 = 104 ,  
uid_1_2_840_10008_5_1_1_30 = 105 ,  
uid_1_2_840_10008_5_1_1_31 = 106 ,  
uid_1_2_840_10008_5_1_1_32 = 107 ,  
uid_1_2_840_10008_5_1_1_33 = 108 ,  
uid_1_2_840_10008_5_1_4_1_1_1 = 109 ,  
uid_1_2_840_10008_5_1_4_1_1_1_1 = 110 ,  
uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111 ,  
uid_1_2_840_10008_5_1_4_1_1_1_2 = 112 ,  
uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113 ,  
uid_1_2_840_10008_5_1_4_1_1_1_3 = 114 ,  
uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115 ,  
uid_1_2_840_10008_5_1_4_1_1_2 = 116 ,
```

[uid_1_2_840_10008_5_1_4_1_1_2_1](#) = 117 ,
[uid_1_2_840_10008_5_1_4_1_1_3](#) = 118 ,
[uid_1_2_840_10008_5_1_4_1_1_3_1](#) = 119 ,
[uid_1_2_840_10008_5_1_4_1_1_4](#) = 120 ,
[uid_1_2_840_10008_5_1_4_1_1_4_1](#) = 121 ,
[uid_1_2_840_10008_5_1_4_1_1_4_2](#) = 122 ,
[uid_1_2_840_10008_5_1_4_1_1_5](#) = 123 ,
[uid_1_2_840_10008_5_1_4_1_1_6](#) = 124 ,
[uid_1_2_840_10008_5_1_4_1_1_6_1](#) = 125 ,
[uid_1_2_840_10008_5_1_4_1_1_7](#) = 126 ,
[uid_1_2_840_10008_5_1_4_1_1_7_1](#) = 127 ,
[uid_1_2_840_10008_5_1_4_1_1_7_2](#) = 128 ,
[uid_1_2_840_10008_5_1_4_1_1_7_3](#) = 129 ,
[uid_1_2_840_10008_5_1_4_1_1_7_4](#) = 130 ,
[uid_1_2_840_10008_5_1_4_1_1_8](#) = 131 ,
[uid_1_2_840_10008_5_1_4_1_1_9](#) = 132 ,
[uid_1_2_840_10008_5_1_4_1_1_9_1](#) = 133 ,
[uid_1_2_840_10008_5_1_4_1_1_9_1_1](#) = 134 ,
[uid_1_2_840_10008_5_1_4_1_1_9_1_2](#) = 135 ,
[uid_1_2_840_10008_5_1_4_1_1_9_1_3](#) = 136 ,
[uid_1_2_840_10008_5_1_4_1_1_9_2_1](#) = 137 ,
[uid_1_2_840_10008_5_1_4_1_1_9_3_1](#) = 138 ,
[uid_1_2_840_10008_5_1_4_1_1_9_4_1](#) = 139 ,
[uid_1_2_840_10008_5_1_4_1_1_10](#) = 140 ,
[uid_1_2_840_10008_5_1_4_1_1_11](#) = 141 ,
[uid_1_2_840_10008_5_1_4_1_1_11_1](#) = 142 ,
[uid_1_2_840_10008_5_1_4_1_1_11_2](#) = 143 ,
[uid_1_2_840_10008_5_1_4_1_1_11_3](#) = 144 ,
[uid_1_2_840_10008_5_1_4_1_1_11_4](#) = 145 ,
[uid_1_2_840_10008_5_1_4_1_1_12_1](#) = 146 ,
[uid_1_2_840_10008_5_1_4_1_1_12_1_1](#) = 147 ,
[uid_1_2_840_10008_5_1_4_1_1_12_2](#) = 148 ,
[uid_1_2_840_10008_5_1_4_1_1_12_2_1](#) = 149 ,
[uid_1_2_840_10008_5_1_4_1_1_13_1_1](#) = 150 ,
[uid_1_2_840_10008_5_1_4_1_1_13_1_2](#) = 151 ,
[uid_1_2_840_10008_5_1_4_1_1_12_3](#) = 152 ,
[uid_1_2_840_10008_5_1_4_1_1_20](#) = 153 ,
[uid_1_2_840_10008_5_1_4_1_1_66](#) = 154 ,
[uid_1_2_840_10008_5_1_4_1_1_66_1](#) = 155 ,
[uid_1_2_840_10008_5_1_4_1_1_66_2](#) = 156 ,
[uid_1_2_840_10008_5_1_4_1_1_66_3](#) = 157 ,
[uid_1_2_840_10008_5_1_4_1_1_66_4](#) = 158 ,
[uid_1_2_840_10008_5_1_4_1_1_67](#) = 159 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1](#) = 160 ,
[uid_1_2_840_10008_5_1_4_1_1_77_2](#) = 161 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_1](#) = 162 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_1_1](#) = 163 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_2](#) = 164 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_2_1](#) = 165 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_3](#) = 166 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_4](#) = 167 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_4_1](#) = 168 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_5_1](#) = 169 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_5_2](#) = 170 ,

```

uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172 ,
uid_1_2_840_10008_5_1_4_1_1_88_1 = 173 ,
uid_1_2_840_10008_5_1_4_1_1_88_2 = 174 ,
uid_1_2_840_10008_5_1_4_1_1_88_3 = 175 ,
uid_1_2_840_10008_5_1_4_1_1_88_4 = 176 ,
uid_1_2_840_10008_5_1_4_1_1_88_11 = 177 ,
uid_1_2_840_10008_5_1_4_1_1_88_22 = 178 ,
uid_1_2_840_10008_5_1_4_1_1_88_33 = 179 ,
uid_1_2_840_10008_5_1_4_1_1_88_40 = 180 ,
uid_1_2_840_10008_5_1_4_1_1_88_50 = 181 ,
uid_1_2_840_10008_5_1_4_1_1_88_59 = 182 ,
uid_1_2_840_10008_5_1_4_1_1_88_65 = 183 ,
uid_1_2_840_10008_5_1_4_1_1_88_67 = 184 ,
uid_1_2_840_10008_5_1_4_1_1_104_1 = 185 ,
uid_1_2_840_10008_5_1_4_1_1_104_2 = 186 ,
uid_1_2_840_10008_5_1_4_1_1_128 = 187 ,
uid_1_2_840_10008_5_1_4_1_1_129 = 188 ,
uid_1_2_840_10008_5_1_4_1_1_481_1 = 189 ,
uid_1_2_840_10008_5_1_4_1_1_481_2 = 190 ,
uid_1_2_840_10008_5_1_4_1_1_481_3 = 191 ,
uid_1_2_840_10008_5_1_4_1_1_481_4 = 192 ,
uid_1_2_840_10008_5_1_4_1_1_481_5 = 193 ,
uid_1_2_840_10008_5_1_4_1_1_481_6 = 194 ,
uid_1_2_840_10008_5_1_4_1_1_481_7 = 195 ,
uid_1_2_840_10008_5_1_4_1_1_481_8 = 196 ,
uid_1_2_840_10008_5_1_4_1_1_481_9 = 197 ,
uid_1_2_840_10008_5_1_4_1_2_1_1 = 198 ,
uid_1_2_840_10008_5_1_4_1_2_1_2 = 199 ,
uid_1_2_840_10008_5_1_4_1_2_1_3 = 200 ,
uid_1_2_840_10008_5_1_4_1_2_2_1 = 201 ,
uid_1_2_840_10008_5_1_4_1_2_2_2 = 202 ,
uid_1_2_840_10008_5_1_4_1_2_2_3 = 203 ,
uid_1_2_840_10008_5_1_4_1_2_3_1 = 204 ,
uid_1_2_840_10008_5_1_4_1_2_3_2 = 205 ,
uid_1_2_840_10008_5_1_4_1_2_3_3 = 206 ,
uid_1_2_840_10008_5_1_4_31 = 207 ,
uid_1_2_840_10008_5_1_4_32_1 = 208 ,
uid_1_2_840_10008_5_1_4_32_2 = 209 ,
uid_1_2_840_10008_5_1_4_32_3 = 210 ,
uid_1_2_840_10008_5_1_4_32 = 211 ,
uid_1_2_840_10008_5_1_4_33 = 212 ,
uid_1_2_840_10008_5_1_4_34_1 = 213 ,
uid_1_2_840_10008_5_1_4_34_2 = 214 ,
uid_1_2_840_10008_5_1_4_34_3 = 215 ,
uid_1_2_840_10008_5_1_4_34_4 = 216 ,
uid_1_2_840_10008_5_1_4_34_4_1 = 217 ,
uid_1_2_840_10008_5_1_4_34_4_2 = 218 ,
uid_1_2_840_10008_5_1_4_34_4_3 = 219 ,
uid_1_2_840_10008_5_1_4_34_4_4 = 220 ,
uid_1_2_840_10008_5_1_4_34_5 = 221 ,
uid_1_2_840_10008_5_1_4_37_1 = 222 ,
uid_1_2_840_10008_5_1_4_37_2 = 223 ,
uid_1_2_840_10008_5_1_4_37_3 = 224 ,

```

[uid_1_2_840_10008_5_1_4_38_1](#) = 225 ,
[uid_1_2_840_10008_5_1_4_38_2](#) = 226 ,
[uid_1_2_840_10008_5_1_4_38_3](#) = 227 ,
[uid_1_2_840_10008_5_1_4_41](#) = 228 ,
[uid_1_2_840_10008_5_1_4_42](#) = 229 ,
[uid_1_2_840_10008_15_0_3_1](#) = 230 ,
[uid_1_2_840_10008_15_0_3_2](#) = 231 ,
[uid_1_2_840_10008_15_0_3_3](#) = 232 ,
[uid_1_2_840_10008_15_0_3_4](#) = 233 ,
[uid_1_2_840_10008_15_0_3_5](#) = 234 ,
[uid_1_2_840_10008_15_0_3_6](#) = 235 ,
[uid_1_2_840_10008_15_0_3_7](#) = 236 ,
[uid_1_2_840_10008_15_0_3_8](#) = 237 ,
[uid_1_2_840_10008_15_0_3_9](#) = 238 ,
[uid_1_2_840_10008_15_0_3_10](#) = 239 ,
[uid_1_2_840_10008_15_0_3_11](#) = 240 ,
[uid_1_2_840_10008_15_0_3_12](#) = 241 ,
[uid_1_2_840_10008_15_0_3_13](#) = 242 ,
[uid_1_2_840_10008_15_0_3_14](#) = 243 ,
[uid_1_2_840_10008_15_0_3_15](#) = 244 ,
[uid_1_2_840_10008_15_0_3_16](#) = 245 ,
[uid_1_2_840_10008_15_0_3_17](#) = 246 ,
[uid_1_2_840_10008_15_0_3_18](#) = 247 ,
[uid_1_2_840_10008_15_0_3_19](#) = 248 ,
[uid_1_2_840_10008_15_0_3_20](#) = 249 ,
[uid_1_2_840_10008_15_0_3_21](#) = 250 ,
[uid_1_2_840_10008_15_0_3_22](#) = 251 ,
[uid_1_2_840_10008_15_0_3_23](#) = 252 ,
[uid_1_2_840_10008_15_0_3_24](#) = 253 ,
[uid_1_2_840_10008_15_0_3_25](#) = 254 ,
[uid_1_2_840_10008_15_0_3_26](#) = 255 ,
[uid_1_2_840_10008_15_0_3_27](#) = 256 ,
[uid_1_2_840_10008_15_0_3_28](#) = 257 ,
[uid_1_2_840_10008_15_0_3_29](#) = 258 ,
[uid_1_2_840_10008_15_0_3_30](#) = 259 ,
[uid_1_2_840_10008_15_0_3_31](#) = 260 ,
[uid_1_2_840_10008_15_0_4_1](#) = 261 ,
[uid_1_2_840_10008_15_0_4_2](#) = 262 ,
[uid_1_2_840_10008_15_0_4_3](#) = 263 ,
[uid_1_2_840_10008_15_0_4_4](#) = 264 ,
[uid_1_2_840_10008_15_0_4_5](#) = 265 ,
[uid_1_2_840_10008_15_0_4_6](#) = 266 ,
[uid_1_2_840_10008_15_0_4_7](#) = 267 ,
[uid_1_2_840_10008_15_0_4_8](#) = 268 ,
[uid_1_2_840_10008_5_1_4_1_1_77_1_6](#) = 269 ,
[uid_1_2_840_10008_5_1_4_1_1_6_2](#) = 270 ,
[uid_1_2_840_10008_5_1_4_1_1_66_5](#) = 271 ,
[uid_1_2_840_10008_5_1_4_1_1_13_1_3](#) = 272 ,
[uid_1_2_840_10008_5_1_4_1_1_2_2](#) = 273 ,
[uid_1_2_840_10008_5_1_4_1_1_4_4](#) = 274 ,
[uid_1_2_840_10008_5_1_4_1_1_128_1](#) = 275 ,
[uid_1_2_840_10008_1_2_4_101](#) = 276 ,
[uid_1_2_840_10008_1_2_4_102](#) = 277 ,
[uid_1_2_840_10008_1_2_4_103](#) = 278 ,

```
uid_1_2_840_10008_1_5_2 = 279 ,
uid_1_2_840_10008_1_5_3 = 280 ,
uid_1_2_840_10008_1_5_4 = 281 ,
uid_1_2_840_10008_1_5_5 = 282 ,
uid_1_2_840_10008_1_5_6 = 283 ,
uid_1_2_840_10008_1_5_7 = 284 ,
uid_1_2_840_10008_1_5_8 = 285 ,
uid_1_2_840_10008_1_20 = 286 ,
uid_1_2_840_10008_2_16_5 = 287 ,
uid_1_2_840_10008_2_16_6 = 288 ,
uid_1_2_840_10008_2_16_7 = 289 ,
uid_1_2_840_10008_2_16_8 = 290 ,
uid_1_2_840_10008_2_16_9 = 291 ,
uid_1_2_840_10008_2_16_10 = 292 ,
uid_1_2_840_10008_2_16_11 = 293 ,
uid_1_2_840_10008_2_16_12 = 294 ,
uid_1_2_840_10008_2_16_13 = 295 ,
uid_1_2_840_10008_2_16_14 = 296 ,
uid_1_2_840_10008_5_1_1_40 = 297 ,
uid_1_2_840_10008_5_1_1_40_1 = 298 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299 ,
uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300 ,
uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301 ,
uid_1_2_840_10008_5_1_4_1_1_11_5 = 302 ,
uid_1_2_840_10008_5_1_4_1_1_11_6 = 303 ,
uid_1_2_840_10008_1_2_4_104 = 304 ,
uid_1_2_840_10008_1_2_4_105 = 305 ,
uid_1_2_840_10008_1_2_4_106 = 306 ,
uid_1_2_840_10008_1_2_4_107 = 307 ,
uid_1_2_840_10008_1_2_4_108 = 308 ,
uid_1_2_840_10008_1_5_1 = 309 ,
uid_1_2_840_10008_5_1_4_1_1_11_7 = 310 ,
uid_1_2_840_10008_5_1_4_1_1_11_8 = 311 ,
uid_1_2_840_10008_5_1_4_1_1_11_9 = 312 ,
uid_1_2_840_10008_5_1_4_1_1_11_10 = 313 ,
uid_1_2_840_10008_5_1_4_1_1_11_11 = 314 ,
uid_1_2_840_10008_5_1_4_1_1_12_77 = 315 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317 ,
uid_1_2_840_10008_5_1_4_1_1_14_1 = 318 ,
uid_1_2_840_10008_5_1_4_1_1_14_2 = 319 ,
uid_1_2_840_10008_5_1_4_1_1_30 = 320 ,
uid_1_2_840_10008_5_1_4_1_1_40 = 321 ,
uid_1_2_840_10008_5_1_4_1_1_66_6 = 322 ,
uid_1_2_840_10008_5_1_4_1_1_68_1 = 323 ,
uid_1_2_840_10008_5_1_4_1_1_68_2 = 324 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328 ,
uid_1_2_840_10008_5_1_4_1_1_78_1 = 329 ,
uid_1_2_840_10008_5_1_4_1_1_78_2 = 330 ,
uid_1_2_840_10008_5_1_4_1_1_78_3 = 331 ,
uid_1_2_840_10008_5_1_4_1_1_78_4 = 332 ,
```


[uid_1_2_840_10008_5_1_4_1_1_78_5](#) = 333 ,
[uid_1_2_840_10008_5_1_4_1_1_78_6](#) = 334 ,
[uid_1_2_840_10008_5_1_4_1_1_78_7](#) = 335 ,
[uid_1_2_840_10008_5_1_4_1_1_78_8](#) = 336 ,
[uid_1_2_840_10008_5_1_4_1_1_79_1](#) = 337 ,
[uid_1_2_840_10008_5_1_4_1_1_80_1](#) = 338 ,
[uid_1_2_840_10008_5_1_4_1_1_81_1](#) = 339 ,
[uid_1_2_840_10008_5_1_4_1_1_82_1](#) = 340 ,
[uid_1_2_840_10008_5_1_4_1_1_88_34](#) = 341 ,
[uid_1_2_840_10008_5_1_4_1_1_88_35](#) = 342 ,
[uid_1_2_840_10008_5_1_4_1_1_88_68](#) = 343 ,
[uid_1_2_840_10008_5_1_4_1_1_88_69](#) = 344 ,
[uid_1_2_840_10008_5_1_4_1_1_88_70](#) = 345 ,
[uid_1_2_840_10008_5_1_4_1_1_88_71](#) = 346 ,
[uid_1_2_840_10008_5_1_4_1_1_88_72](#) = 347 ,
[uid_1_2_840_10008_5_1_4_1_1_88_73](#) = 348 ,
[uid_1_2_840_10008_5_1_4_1_1_88_74](#) = 349 ,
[uid_1_2_840_10008_5_1_4_1_1_88_75](#) = 350 ,
[uid_1_2_840_10008_5_1_4_1_1_90_1](#) = 351 ,
[uid_1_2_840_10008_5_1_4_1_1_104_3](#) = 352 ,
[uid_1_2_840_10008_5_1_4_1_1_130](#) = 353 ,
[uid_1_2_840_10008_5_1_4_1_1_131](#) = 354 ,
[uid_1_2_840_10008_5_1_4_1_1_200_1](#) = 355 ,
[uid_1_2_840_10008_5_1_4_1_1_200_2](#) = 356 ,
[uid_1_2_840_10008_5_1_4_1_1_200_3](#) = 357 ,
[uid_1_2_840_10008_5_1_4_1_1_200_4](#) = 358 ,
[uid_1_2_840_10008_5_1_4_1_1_200_5](#) = 359 ,
[uid_1_2_840_10008_5_1_4_1_1_200_6](#) = 360 ,
[uid_1_2_840_10008_5_1_4_1_1_481_10](#) = 361 ,
[uid_1_2_840_10008_5_1_4_1_1_481_11](#) = 362 ,
[uid_1_2_840_10008_5_1_4_1_1_501_1](#) = 363 ,
[uid_1_2_840_10008_5_1_4_1_1_501_2_1](#) = 364 ,
[uid_1_2_840_10008_5_1_4_1_1_501_2_2](#) = 365 ,
[uid_1_2_840_10008_5_1_4_1_1_501_3](#) = 366 ,
[uid_1_2_840_10008_5_1_4_1_1_501_4](#) = 367 ,
[uid_1_2_840_10008_5_1_4_1_1_501_5](#) = 368 ,
[uid_1_2_840_10008_5_1_4_1_1_501_6](#) = 369 ,
[uid_1_2_840_10008_5_1_4_1_1_601_1](#) = 370 ,
[uid_1_2_840_10008_5_1_4_1_1_601_2](#) = 371 ,
[uid_1_2_840_10008_5_1_4_1_2_4_2](#) = 372 ,
[uid_1_2_840_10008_5_1_4_1_2_4_3](#) = 373 ,
[uid_1_2_840_10008_5_1_4_1_2_5_3](#) = 374 ,
[uid_1_2_840_10008_5_1_4_20_1](#) = 375 ,
[uid_1_2_840_10008_5_1_4_20_2](#) = 376 ,
[uid_1_2_840_10008_5_1_4_20_3](#) = 377 ,
[uid_1_2_840_10008_5_1_4_34_5_1](#) = 378 ,
[uid_1_2_840_10008_5_1_4_34_6](#) = 379 ,
[uid_1_2_840_10008_5_1_4_34_6_1](#) = 380 ,
[uid_1_2_840_10008_5_1_4_34_6_2](#) = 381 ,
[uid_1_2_840_10008_5_1_4_34_6_3](#) = 382 ,
[uid_1_2_840_10008_5_1_4_34_6_4](#) = 383 ,
[uid_1_2_840_10008_5_1_4_34_7](#) = 384 ,
[uid_1_2_840_10008_5_1_4_34_8](#) = 385 ,
[uid_1_2_840_10008_5_1_4_34_9](#) = 386 ,

```

uid_1_2_840_10008_5_1_4_34_10 = 387 ,
uid_1_2_840_10008_5_1_4_38_4 = 388 ,
uid_1_2_840_10008_5_1_4_39_1 = 389 ,
uid_1_2_840_10008_5_1_4_39_2 = 390 ,
uid_1_2_840_10008_5_1_4_39_3 = 391 ,
uid_1_2_840_10008_5_1_4_39_4 = 392 ,
uid_1_2_840_10008_5_1_4_43_1 = 393 ,
uid_1_2_840_10008_5_1_4_43_2 = 394 ,
uid_1_2_840_10008_5_1_4_43_3 = 395 ,
uid_1_2_840_10008_5_1_4_43_4 = 396 ,
uid_1_2_840_10008_5_1_4_44_1 = 397 ,
uid_1_2_840_10008_5_1_4_44_2 = 398 ,
uid_1_2_840_10008_5_1_4_44_3 = 399 ,
uid_1_2_840_10008_5_1_4_44_4 = 400 ,
uid_1_2_840_10008_5_1_4_45_1 = 401 ,
uid_1_2_840_10008_5_1_4_45_2 = 402 ,
uid_1_2_840_10008_5_1_4_45_3 = 403 ,
uid_1_2_840_10008_5_1_4_45_4 = 404 ,
uid_1_2_840_10008_7_1_1 = 405 ,
uid_1_2_840_10008_7_1_2 = 406 ,
uid_1_2_840_10008_8_1_1 = 407 ,
uid_1_2_840_10008_5_1_4_1_1_4_3 = 408 ,
uid_1_2_840_10008_15_1_1 = 409 }

```

Public Member Functions

- [UIDs](#) ()=default
- const char * [GetName](#) () const
- const char * [GetString](#) () const
- [operator TSType](#) () const
- bool [SetFromUID](#) (const char *str)

Static Public Member Functions

- static unsigned int [GetNumberOfTransferSyntaxStrings](#) ()
- static const char *const * [GetTransferSyntaxString](#) (unsigned int ts)
- static [TransferSyntaxStringsType](#) [GetTransferSyntaxStrings](#) ()
- static const char * [GetUIDName](#) (unsigned int ts)
- static const char * [GetUIDString](#) (unsigned int ts)

12.324.1 Detailed Description

all known uids

Examples

[GenerateStandardSOPClasses.cxx](#).

12.324.2 Member Typedef Documentation

12.324.2.1 TransferSyntaxStringsType

```
typedef const char* const(* gdcmm::UIDs::TransferSyntaxStringsType)[2]
```

12.324.3 Member Enumeration Documentation

12.324.3.1 TSName

```
enum gdcmm::UIDs::TSName
```

Enumerator

VerificationSOPClass
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
ExplicitVRLittleEndian
DeflatedExplicitVRLittleEndian
ExplicitVRBigEndian
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only
JPEGExtendedProcess35Retired
JPEGSpectralSelectionNonHierarchicalProcess68Retired
JPEGSpectralSelectionNonHierarchicalProcess79Retired
JPEGFullProgressionNonHierarchicalProcess1012Retired
JPEGFullProgressionNonHierarchicalProcess1113Retired
JPEGLosslessNonHierarchicalProcess14
JPEGLosslessNonHierarchicalProcess15Retired
JPEGExtendedHierarchicalProcess1618Retired
JPEGExtendedHierarchicalProcess1719Retired
JPEGSpectralSelectionHierarchicalProcess2022Retired
JPEGSpectralSelectionHierarchicalProcess2123Retired
JPEGFullProgressionHierarchicalProcess2426Retired
JPEGFullProgressionHierarchicalProcess2527Retired
JPEGLosslessHierarchicalProcess28Retired
JPEGLosslessHierarchicalProcess29Retired
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImage
JPEGLSLosslessImageCompression
JPEGLSLossyNearLosslessImageCompression
JPEG2000ImageCompressionLosslessOnly

JPEG2000ImageCompression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
JPEG2000Part2MulticomponentImageCompression
JPIPReferenced
JPIPReferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFFrameofReference
SPM2EPIFrameofReference
SPM2FILT1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference
SPM2AVG152T1FrameofReference
SPM2AVG152T2FrameofReference
SPM2AVG152PDFFrameofReference
SPM2SINGLESUBJT1FrameofReference
ICBM452T1FrameofReference
ICBMSingleSubjectMRIFrameofReference
BasicStudyContentNotificationSOPClassRetired
StorageCommitmentPushModelSOPClass
StorageCommitmentPushModelSOPInstance
StorageCommitmentPullModelSOPClassRetired
StorageCommitmentPullModelSOPInstanceRetired
ProceduralEventLoggingSOPClass
ProceduralEventLoggingSOPInstance
SubstanceAdministrationLoggingSOPClass
SubstanceAdministrationLoggingSOPInstance

DICOMUIDRegistry
DICOMControlledTerminology
DICOMApplicationContextName
DetachedPatientManagementSOPClassRetired
DetachedPatientManagementMetaSOPClassRetired
DetachedVisitManagementSOPClassRetired
DetachedStudyManagementSOPClassRetired
StudyComponentManagementSOPClassRetired
ModalityPerformedProcedureStepSOPClass
ModalityPerformedProcedureStepRetrieveSOPClass
ModalityPerformedProcedureStepNotificationSOPClass
DetachedResultsManagementSOPClassRetired
DetachedResultsManagementMetaSOPClassRetired
DetachedStudyManagementMetaSOPClassRetired
DetachedInterpretationManagementSOPClassRetired
StorageServiceClass
BasicFilmSessionSOPClass
BasicFilmBoxSOPClass
BasicGrayscaleImageBoxSOPClass
BasicColorImageBoxSOPClass
ReferencedImageBoxSOPClassRetired
BasicGrayscalePrintManagementMetaSOPClass
ReferencedGrayscalePrintManagementMetaSOPClassRetired
PrintJobSOPClass
BasicAnnotationBoxSOPClass
PrinterSOPClass
PrinterConfigurationRetrievalSOPClass
PrinterSOPInstance
PrinterConfigurationRetrievalSOPInstance
BasicColorPrintManagementMetaSOPClass
ReferencedColorPrintManagementMetaSOPClassRetired
VOILUTBoxSOPClass
PresentationLUTSOPClass
ImageOverlayBoxSOPClassRetired
BasicPrintImageOverlayBoxSOPClassRetired
PrintQueueSOPInstanceRetired
PrintQueueManagementSOPClassRetired
StoredPrintStorageSOPClassRetired
HardcopyGrayscaleImageStorageSOPClassRetired

HardcopyColorImageStorageSOPClassRetired
PullPrintRequestSOPClassRetired
PullStoredPrintManagementMetaSOPClassRetired
MediaCreationManagementSOPClassUID
ComputedRadiographyImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographyXRayImageStorageForPresentation
DigitalMammographyXRayImageStorageForProcessing
DigitalIntraoralXRayImageStorageForPresentation
DigitalIntraoralXRayImageStorageForProcessing
CTImageStorage
EnhancedCTImageStorage
UltrasoundMultiframeImageStorageRetired
UltrasoundMultiframeImageStorage
MRImageStorage
EnhancedMRImageStorage
MRSpectroscopyStorage
NuclearMedicineImageStorageRetired
UltrasoundImageStorageRetired
UltrasoundImageStorage
SecondaryCaptureImageStorage
MultiframeSingleBitSecondaryCaptureImageStorage
MultiframeGrayscaleByteSecondaryCaptureImageStorage
MultiframeGrayscaleWordSecondaryCaptureImageStorage
MultiframeTrueColorSecondaryCaptureImageStorage
StandaloneOverlayStorageRetired
StandaloneCurveStorageRetired
WaveformStorageTrialRetired
ECG12leadWaveformStorage
GeneralECGWaveformStorage
AmbulatoryECGWaveformStorage
HemodynamicWaveformStorage
CardiacElectrophysiologyWaveformStorage
BasicVoiceAudioWaveformStorage
StandaloneModalityLUTStorageRetired
StandaloneVOILUTStorageRetired
GrayscaleSoftcopyPresentationStateStorageSOPClass
ColorSoftcopyPresentationStateStorageSOPClass

PseudoColorSoftcopyPresentationStateStorageSOPClass
BlendingSoftcopyPresentationStateStorageSOPClass
XRayAngiographicImageStorage
EnhancedXAImageStorage
XRayRadiofluoroscopicImageStorage
EnhancedXRFImageStorage
XRay3DAngiographicImageStorage
XRay3DCraniofacialImageStorage
XRayAngiographicBiPlaneImageStorageRetired
NuclearMedicineImageStorage
RawDataStorage
SpatialRegistrationStorage
SpatialFiducialsStorage
DeformableSpatialRegistrationStorage
SegmentationStorage
RealWorldValueMappingStorage
VLImageStorageTrialRetired
VLMultiframeImageStorageTrialRetired
VLEndoscopicImageStorage
VideoEndoscopicImageStorage
VLMicroscopicImageStorage
VideoMicroscopicImageStorage
VLSlideCoordinatesMicroscopicImageStorage
VLPhotographicImageStorage
VideoPhotographicImageStorage
OphthalmicPhotography8BitImageStorage
OphthalmicPhotography16BitImageStorage
StereometricRelationshipStorage
OphthalmicTomographyImageStorage
TextSRStorageTrialRetired
AudioSRStorageTrialRetired
DetailSRStorageTrialRetired
ComprehensiveSRStorageTrialRetired
BasicTextSRStorage
EnhancedSRStorage
ComprehensiveSRStorage
ProcedureLogStorage
MammographyCADSRStorage
KeyObjectSelectionDocumentStorage

ChestCADSRStorage
XRayRadiationDoseSRStorage
EncapsulatedPDFStorage
EncapsulatedCDASStorage
PositronEmissionTomographyImageStorage
StandalonePETCurveStorageRetired
RTImageStorage
RTDoseStorage
RTStructureSetStorage
RTBeamsTreatmentRecordStorage
RTPlanStorage
RTBrachyTreatmentRecordStorage
RTTreatmentSummaryRecordStorage
RTIonPlanStorage
RTIonBeamsTreatmentRecordStorage
PatientRootQueryRetrieveInformationModelFIND
PatientRootQueryRetrieveInformationModelMOVE
PatientRootQueryRetrieveInformationModelGET
StudyRootQueryRetrieveInformationModelFIND
StudyRootQueryRetrieveInformationModelMOVE
StudyRootQueryRetrieveInformationModelGET
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
PatientStudyOnlyQueryRetrieveInformationModelGETRetired
ModalityWorklistInformationModelFIND
GeneralPurposeWorklistInformationModelFIND
GeneralPurposeScheduledProcedureStepSOPClass
GeneralPurposePerformedProcedureStepSOPClass
GeneralPurposeWorklistManagementMetaSOPClass
InstanceAvailabilityNotificationSOPClass
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
RTConventionalMachineVerificationSupplement74FrozenDraft
RTIonMachineVerificationSupplement74FrozenDraft
UnifiedWorklistandProcedureStepServiceClass
UnifiedProcedureStepPushSOPClass
UnifiedProcedureStepWatchSOPClass
UnifiedProcedureStepPullSOPClass
UnifiedProcedureStepEventSOPClass
UnifiedWorklistandProcedureStepSOPInstance

GeneralRelevantPatientInformationQuery
BreastImagingRelevantPatientInformationQuery
CardiacRelevantPatientInformationQuery
HangingProtocolStorage
HangingProtocolInformationModelFIND
HangingProtocolInformationModelMOVE
ProductCharacteristicsQuerySOPClass
SubstanceApprovalQuerySOPClass
dicomDeviceName
dicomDescription
dicomManufacturer
dicomManufacturerModelName
dicomSoftwareVersion
dicomVendorData
dicomAETitle
dicomNetworkConnectionReference
dicomApplicationCluster
dicomAssociationInitiator
dicomAssociationAcceptor
dicomHostname
dicomPort
dicomSOPClass
dicomTransferRole
dicomTransferSyntax
dicomPrimaryDeviceType
dicomRelatedDeviceReference
dicomPreferredCalledAETitle
dicomTLSCyphersuite
dicomAuthorizedNodeCertificateReference
dicomThisNodeCertificateReference
dicomInstalled
dicomStationName
dicomDeviceSerialNumber
dicomInstitutionName
dicomInstitutionAddress
dicomInstitutionDepartmentName
dicomIssuerOfPatientID
dicomPreferredCallingAETitle
dicomSupportedCharacterSet

dicomConfigurationRoot
dicomDevicesRoot
dicomUniqueAETitlesRegistryRoot
dicomDevice
dicomNetworkAE
dicomNetworkConnection
dicomUniqueAETitle
dicomTransferCapability
VLWholeSlideMicroscopyImageStorage
EnhancedUSVolumeStorage
SurfaceSegmentationStorage
BreastTomosynthesisImageStorage
LegacyConvertedEnhancedCTImageStorage
LegacyConvertedEnhancedMRImageStorage
LegacyConvertedEnhancedPETImageStorage
MPEG2MainProfileHighLevel
MPEG4AVCH_264HighProfileLevel4_1
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1
PETColorPaletteSOPInstance
HotMetalBlueColorPaletteSOPInstance
PET20StepColorPaletteSOPInstance
SpringColorPaletteSOPInstance
SummerColorPaletteSOPInstance
FallColorPaletteSOPInstance
WinterColorPaletteSOPInstance
Papyrus3ImplicitVRLittleEndian
AdultMouseAnatomyOntology
UberonOntology
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN
MouseGenomeInitiativeMGI
PubChemCompoundCID
ICD11
NewYorkUniversityMelanomaClinicalCooperativeGroup
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide
ImageBiomarkerStandardisationInitiative
RadiomicsOntology
DisplaySystemSOPClass
DisplaySystemSOPInstance

GeneralAudioWaveformStorage
ArterialPulseWaveformStorage
RespiratoryWaveformStorage
XAXRFGayscaleSoftcopyPresentationStateStorage
GrayscalePlanarMPRVolumetricPresentationStateStorage
MPEG4AVCH_264HighProfileLevel4_2For2DVideo
MPEG4AVCH_264HighProfileLevel4_2For3DVideo
MPEG4AVCH_264StereoHighProfileLevel4_2
HEVCH_265MainProfileLevel5_1
HEVCH_265Main10ProfileLevel5_1
HotIronColorPaletteSOPInstance
CompositingPlanarMPRVolumetricPresentationStateStorage
AdvancedBlendingPresentationStateStorage
VolumeRenderingVolumetricPresentationStateStorage
SegmentedVolumeRenderingVolumetricPresentationStateStorage
MultipleVolumeRenderingVolumetricPresentationStateStorage
Null0
BreastProjectionXRayImageStorageForPresentation
BreastProjectionXRayImageStorageForProcessing
IntravascularOpticalCoherenceTomographyImageStorageForPresentation
IntravascularOpticalCoherenceTomographyImageStorageForProcessing
ParametricMapStorage
Null1
TractographyResultsStorage
SurfaceScanMeshStorage
SurfaceScanPointCloudStorage
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
WideFieldOphthalmicPhotography3DCoordinatesImageStorage
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
LensometryMeasurementsStorage
AutorefractionMeasurementsStorage
KeratometryMeasurementsStorage
SubjectiveRefractionMeasurementsStorage
VisualAcuityMeasurementsStorage
SpectaclePrescriptionReportStorage
OphthalmicAxialMeasurementsStorage

IntraocularLensCalculationsStorage
MacularGridThicknessandVolumeReportStorage
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage
OphthalmicThicknessMapStorage
CornealTopographyMapStorage
Comprehensive3DSRStorage
ExtensibleSRStorage
RadiopharmaceuticalRadiationDoseSRStorage
ColonCADSRStorage
ImplantationPlanSRStorage
AcquisitionContextSRStorage
SimplifiedAdultEchoSRStorage
PatientRadiationDoseSRStorage
PlannedImagingAgentAdministrationSRStorage
PerformedImagingAgentAdministrationSRStorage
ContentAssessmentResultsStorage
EncapsulatedSTLStorage
EnhancedPETImageStorage
BasicStructuredDisplayStorage
CTDefinedProcedureProtocolStorage
CTPerformedProcedureProtocolStorage
ProtocolApprovalStorage
ProtocolApprovalInformationModelFIND
ProtocolApprovalInformationModelMOVE
ProtocolApprovalInformationModelGET
RTPhysicianIntentStorage
RTSegmentAnnotationStorage
DICOSCTImageStorage
DICOSDigitalXRayImageStorageForPresentation
DICOSDigitalXRayImageStorageForProcessing
DICOSThreatDetectionReportStorage
DICOS2DAITStorage
DICOS3DAITStorage
DICOSQuadrupoleResonanceQRStorage
EddyCurrentImageStorage
EddyCurrentMultiframeImageStorage
CompositeInstanceRootRetrieveMOVE
CompositeInstanceRootRetrieveGET
CompositeInstanceRetrieveWithoutBulkDataGET

DefinedProcedureProtocolInformationModelFIND
DefinedProcedureProtocolInformationModelMOVE
DefinedProcedureProtocolInformationModelGET
UPSFilteredGlobalSubscriptionSOPInstance
UnifiedWorklistandProcedureStepServiceClass1
UnifiedProcedureStepPushSOPClass1
UnifiedProcedureStepWatchSOPClass1
UnifiedProcedureStepPullSOPClass1
UnifiedProcedureStepEventSOPClass1
RTBeamsDeliveryInstructionStorage
RTConventionalMachineVerification
RTIonMachineVerification
RTBrachyApplicationSetupDeliveryInstructionStorage
HangingProtocolInformationModelGET
ColorPaletteStorage
ColorPaletteQueryRetrieveInformationModelFIND
ColorPaletteQueryRetrieveInformationModelMOVE
ColorPaletteQueryRetrieveInformationModelGET
GenericImplantTemplateStorage
GenericImplantTemplateInformationModelFIND
GenericImplantTemplateInformationModelMOVE
GenericImplantTemplateInformationModelGET
ImplantAssemblyTemplateStorage
ImplantAssemblyTemplateInformationModelFIND
ImplantAssemblyTemplateInformationModelMOVE
ImplantAssemblyTemplateInformationModelGET
ImplantTemplateGroupStorage
ImplantTemplateGroupInformationModelFIND
ImplantTemplateGroupInformationModelMOVE
ImplantTemplateGroupInformationModelGET
NativeDICOMModel
AbstractMultiDimensionalImageModel
DICOMContentMappingResource
EnhancedMRColorImageStorage
UniversalCoordinatedTime

12.324.3.2 TSType

enum [gdcmm::UIDs::TSType](#)

Enumerator

uid_1_2_840_10008_1_1	
uid_1_2_840_10008_1_2	
uid_1_2_840_10008_1_2_1	
uid_1_2_840_10008_1_2_1_99	
uid_1_2_840_10008_1_2_2	
uid_1_2_840_10008_1_2_4_50	
uid_1_2_840_10008_1_2_4_51	
uid_1_2_840_10008_1_2_4_52	
uid_1_2_840_10008_1_2_4_53	
uid_1_2_840_10008_1_2_4_54	
uid_1_2_840_10008_1_2_4_55	
uid_1_2_840_10008_1_2_4_56	
uid_1_2_840_10008_1_2_4_57	
uid_1_2_840_10008_1_2_4_58	
uid_1_2_840_10008_1_2_4_59	
uid_1_2_840_10008_1_2_4_60	
uid_1_2_840_10008_1_2_4_61	
uid_1_2_840_10008_1_2_4_62	
uid_1_2_840_10008_1_2_4_63	
uid_1_2_840_10008_1_2_4_64	
uid_1_2_840_10008_1_2_4_65	
uid_1_2_840_10008_1_2_4_66	
uid_1_2_840_10008_1_2_4_70	
uid_1_2_840_10008_1_2_4_80	
uid_1_2_840_10008_1_2_4_81	
uid_1_2_840_10008_1_2_4_90	
uid_1_2_840_10008_1_2_4_91	
uid_1_2_840_10008_1_2_4_92	
uid_1_2_840_10008_1_2_4_93	
uid_1_2_840_10008_1_2_4_94	
uid_1_2_840_10008_1_2_4_95	
uid_1_2_840_10008_1_2_4_100	
uid_1_2_840_10008_1_2_5	
uid_1_2_840_10008_1_2_6_1	

uid_1_2_840_10008_1_2_6_2	
uid_1_2_840_10008_1_3_10	
uid_1_2_840_10008_1_4_1_1	
uid_1_2_840_10008_1_4_1_2	
uid_1_2_840_10008_1_4_1_3	
uid_1_2_840_10008_1_4_1_4	
uid_1_2_840_10008_1_4_1_5	
uid_1_2_840_10008_1_4_1_6	
uid_1_2_840_10008_1_4_1_7	
uid_1_2_840_10008_1_4_1_8	
uid_1_2_840_10008_1_4_1_9	
uid_1_2_840_10008_1_4_1_10	
uid_1_2_840_10008_1_4_1_11	
uid_1_2_840_10008_1_4_1_12	
uid_1_2_840_10008_1_4_1_13	
uid_1_2_840_10008_1_4_1_14	
uid_1_2_840_10008_1_4_1_15	
uid_1_2_840_10008_1_4_1_16	
uid_1_2_840_10008_1_4_1_17	
uid_1_2_840_10008_1_4_1_18	
uid_1_2_840_10008_1_4_2_1	
uid_1_2_840_10008_1_4_2_2	
uid_1_2_840_10008_1_9	
uid_1_2_840_10008_1_20_1	
uid_1_2_840_10008_1_20_1_1	
uid_1_2_840_10008_1_20_2	
uid_1_2_840_10008_1_20_2_1	

uid_1_2_840_10008_1_40	
uid_1_2_840_10008_1_40_1	
uid_1_2_840_10008_1_42	
uid_1_2_840_10008_1_42_1	
uid_1_2_840_10008_2_6_1	
uid_1_2_840_10008_2_16_4	
uid_1_2_840_10008_3_1_1_1	
uid_1_2_840_10008_3_1_2_1_1	
uid_1_2_840_10008_3_1_2_1_4	
uid_1_2_840_10008_3_1_2_2_1	
uid_1_2_840_10008_3_1_2_3_1	
uid_1_2_840_10008_3_1_2_3_2	
uid_1_2_840_10008_3_1_2_3_3	
uid_1_2_840_10008_3_1_2_3_4	
uid_1_2_840_10008_3_1_2_3_5	
uid_1_2_840_10008_3_1_2_5_1	
uid_1_2_840_10008_3_1_2_5_4	
uid_1_2_840_10008_3_1_2_5_5	
uid_1_2_840_10008_3_1_2_6_1	
uid_1_2_840_10008_4_2	
uid_1_2_840_10008_5_1_1_1	
uid_1_2_840_10008_5_1_1_2	
uid_1_2_840_10008_5_1_1_4	

uid_1_2_840_10008_5_1_1_4_1	
uid_1_2_840_10008_5_1_1_4_2	
uid_1_2_840_10008_5_1_1_9	
uid_1_2_840_10008_5_1_1_9_1	
uid_1_2_840_10008_5_1_1_14	
uid_1_2_840_10008_5_1_1_15	
uid_1_2_840_10008_5_1_1_16	
uid_1_2_840_10008_5_1_1_16_376	
uid_1_2_840_10008_5_1_1_17	
uid_1_2_840_10008_5_1_1_17_376	
uid_1_2_840_10008_5_1_1_18	
uid_1_2_840_10008_5_1_1_18_1	
uid_1_2_840_10008_5_1_1_22	
uid_1_2_840_10008_5_1_1_23	
uid_1_2_840_10008_5_1_1_24	
uid_1_2_840_10008_5_1_1_24_1	
uid_1_2_840_10008_5_1_1_25	
uid_1_2_840_10008_5_1_1_26	
uid_1_2_840_10008_5_1_1_27	
uid_1_2_840_10008_5_1_1_29	
uid_1_2_840_10008_5_1_1_30	
uid_1_2_840_10008_5_1_1_31	
uid_1_2_840_10008_5_1_1_32	
uid_1_2_840_10008_5_1_1_33	
uid_1_2_840_10008_5_1_4_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_1_3	

uid_1_2_840_10008_5_1_4_1_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_2	
uid_1_2_840_10008_5_1_4_1_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_3	
uid_1_2_840_10008_5_1_4_1_1_3_1	
uid_1_2_840_10008_5_1_4_1_1_4	
uid_1_2_840_10008_5_1_4_1_1_4_1	
uid_1_2_840_10008_5_1_4_1_1_4_2	
uid_1_2_840_10008_5_1_4_1_1_5	
uid_1_2_840_10008_5_1_4_1_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_1	
uid_1_2_840_10008_5_1_4_1_1_7	
uid_1_2_840_10008_5_1_4_1_1_7_1	
uid_1_2_840_10008_5_1_4_1_1_7_2	
uid_1_2_840_10008_5_1_4_1_1_7_3	
uid_1_2_840_10008_5_1_4_1_1_7_4	
uid_1_2_840_10008_5_1_4_1_1_8	
uid_1_2_840_10008_5_1_4_1_1_9	
uid_1_2_840_10008_5_1_4_1_1_9_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_1	
uid_1_2_840_10008_5_1_4_1_1_9_1_2	
uid_1_2_840_10008_5_1_4_1_1_9_1_3	

uid_1_2_840_10008_5_1_4_1_1_9_2_1	
uid_1_2_840_10008_5_1_4_1_1_9_3_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_1	
uid_1_2_840_10008_5_1_4_1_1_10	
uid_1_2_840_10008_5_1_4_1_1_11	
uid_1_2_840_10008_5_1_4_1_1_11_1	
uid_1_2_840_10008_5_1_4_1_1_11_2	
uid_1_2_840_10008_5_1_4_1_1_11_3	
uid_1_2_840_10008_5_1_4_1_1_11_4	
uid_1_2_840_10008_5_1_4_1_1_12_1	
uid_1_2_840_10008_5_1_4_1_1_12_1_1	
uid_1_2_840_10008_5_1_4_1_1_12_2	
uid_1_2_840_10008_5_1_4_1_1_12_2_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_1	
uid_1_2_840_10008_5_1_4_1_1_13_1_2	
uid_1_2_840_10008_5_1_4_1_1_12_3	
uid_1_2_840_10008_5_1_4_1_1_20	
uid_1_2_840_10008_5_1_4_1_1_66	
uid_1_2_840_10008_5_1_4_1_1_66_1	
uid_1_2_840_10008_5_1_4_1_1_66_2	
uid_1_2_840_10008_5_1_4_1_1_66_3	
uid_1_2_840_10008_5_1_4_1_1_66_4	
uid_1_2_840_10008_5_1_4_1_1_67	
uid_1_2_840_10008_5_1_4_1_1_77_1	
uid_1_2_840_10008_5_1_4_1_1_77_2	

uid_1_2_840_10008_5_1_4_1_1_77_1_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_3	
uid_1_2_840_10008_5_1_4_1_1_77_1_4	
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4	
uid_1_2_840_10008_5_1_4_1_1_88_1	
uid_1_2_840_10008_5_1_4_1_1_88_2	
uid_1_2_840_10008_5_1_4_1_1_88_3	
uid_1_2_840_10008_5_1_4_1_1_88_4	
uid_1_2_840_10008_5_1_4_1_1_88_11	
uid_1_2_840_10008_5_1_4_1_1_88_22	
uid_1_2_840_10008_5_1_4_1_1_88_33	
uid_1_2_840_10008_5_1_4_1_1_88_40	
uid_1_2_840_10008_5_1_4_1_1_88_50	
uid_1_2_840_10008_5_1_4_1_1_88_59	
uid_1_2_840_10008_5_1_4_1_1_88_65	
uid_1_2_840_10008_5_1_4_1_1_88_67	
uid_1_2_840_10008_5_1_4_1_1_104_1	
uid_1_2_840_10008_5_1_4_1_1_104_2	
uid_1_2_840_10008_5_1_4_1_1_128	
uid_1_2_840_10008_5_1_4_1_1_129	

uid_1_2_840_10008_5_1_4_1_1_481_1	
uid_1_2_840_10008_5_1_4_1_1_481_2	
uid_1_2_840_10008_5_1_4_1_1_481_3	
uid_1_2_840_10008_5_1_4_1_1_481_4	
uid_1_2_840_10008_5_1_4_1_1_481_5	
uid_1_2_840_10008_5_1_4_1_1_481_6	
uid_1_2_840_10008_5_1_4_1_1_481_7	
uid_1_2_840_10008_5_1_4_1_1_481_8	
uid_1_2_840_10008_5_1_4_1_1_481_9	
uid_1_2_840_10008_5_1_4_1_2_1_1	
uid_1_2_840_10008_5_1_4_1_2_1_2	
uid_1_2_840_10008_5_1_4_1_2_1_3	
uid_1_2_840_10008_5_1_4_1_2_2_1	
uid_1_2_840_10008_5_1_4_1_2_2_2	
uid_1_2_840_10008_5_1_4_1_2_2_3	
uid_1_2_840_10008_5_1_4_1_2_3_1	
uid_1_2_840_10008_5_1_4_1_2_3_2	
uid_1_2_840_10008_5_1_4_1_2_3_3	
uid_1_2_840_10008_5_1_4_31	
uid_1_2_840_10008_5_1_4_32_1	
uid_1_2_840_10008_5_1_4_32_2	
uid_1_2_840_10008_5_1_4_32_3	
uid_1_2_840_10008_5_1_4_32	
uid_1_2_840_10008_5_1_4_33	

uid_1_2_840_10008_5_1_4_34_1	
uid_1_2_840_10008_5_1_4_34_2	
uid_1_2_840_10008_5_1_4_34_3	
uid_1_2_840_10008_5_1_4_34_4	
uid_1_2_840_10008_5_1_4_34_4_1	
uid_1_2_840_10008_5_1_4_34_4_2	
uid_1_2_840_10008_5_1_4_34_4_3	
uid_1_2_840_10008_5_1_4_34_4_4	
uid_1_2_840_10008_5_1_4_34_5	
uid_1_2_840_10008_5_1_4_37_1	
uid_1_2_840_10008_5_1_4_37_2	
uid_1_2_840_10008_5_1_4_37_3	
uid_1_2_840_10008_5_1_4_38_1	
uid_1_2_840_10008_5_1_4_38_2	
uid_1_2_840_10008_5_1_4_38_3	
uid_1_2_840_10008_5_1_4_41	
uid_1_2_840_10008_5_1_4_42	
uid_1_2_840_10008_15_0_3_1	
uid_1_2_840_10008_15_0_3_2	
uid_1_2_840_10008_15_0_3_3	
uid_1_2_840_10008_15_0_3_4	
uid_1_2_840_10008_15_0_3_5	
uid_1_2_840_10008_15_0_3_6	

uid_1_2_840_10008_15_0_3_7	
uid_1_2_840_10008_15_0_3_8	
uid_1_2_840_10008_15_0_3_9	
uid_1_2_840_10008_15_0_3_10	
uid_1_2_840_10008_15_0_3_11	
uid_1_2_840_10008_15_0_3_12	
uid_1_2_840_10008_15_0_3_13	
uid_1_2_840_10008_15_0_3_14	
uid_1_2_840_10008_15_0_3_15	
uid_1_2_840_10008_15_0_3_16	
uid_1_2_840_10008_15_0_3_17	
uid_1_2_840_10008_15_0_3_18	
uid_1_2_840_10008_15_0_3_19	
uid_1_2_840_10008_15_0_3_20	
uid_1_2_840_10008_15_0_3_21	
uid_1_2_840_10008_15_0_3_22	
uid_1_2_840_10008_15_0_3_23	
uid_1_2_840_10008_15_0_3_24	
uid_1_2_840_10008_15_0_3_25	
uid_1_2_840_10008_15_0_3_26	
uid_1_2_840_10008_15_0_3_27	
uid_1_2_840_10008_15_0_3_28	
uid_1_2_840_10008_15_0_3_29	
uid_1_2_840_10008_15_0_3_30	
uid_1_2_840_10008_15_0_3_31	
uid_1_2_840_10008_15_0_4_1	
uid_1_2_840_10008_15_0_4_2	
uid_1_2_840_10008_15_0_4_3	
uid_1_2_840_10008_15_0_4_4	
uid_1_2_840_10008_15_0_4_5	
uid_1_2_840_10008_15_0_4_6	
uid_1_2_840_10008_15_0_4_7	

uid_1_2_840_10008_15_0_4_8	
uid_1_2_840_10008_5_1_4_1_1_77_1_6	
uid_1_2_840_10008_5_1_4_1_1_6_2	
uid_1_2_840_10008_5_1_4_1_1_66_5	
uid_1_2_840_10008_5_1_4_1_1_13_1_3	
uid_1_2_840_10008_5_1_4_1_1_2_2	
uid_1_2_840_10008_5_1_4_1_1_4_4	
uid_1_2_840_10008_5_1_4_1_1_128_1	
uid_1_2_840_10008_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	
uid_1_2_840_10008_1_2_4_103	
uid_1_2_840_10008_1_5_2	
uid_1_2_840_10008_1_5_3	
uid_1_2_840_10008_1_5_4	
uid_1_2_840_10008_1_5_5	
uid_1_2_840_10008_1_5_6	
uid_1_2_840_10008_1_5_7	
uid_1_2_840_10008_1_5_8	
uid_1_2_840_10008_1_20	
uid_1_2_840_10008_2_16_5	
uid_1_2_840_10008_2_16_6	
uid_1_2_840_10008_2_16_7	
uid_1_2_840_10008_2_16_8	
uid_1_2_840_10008_2_16_9	
uid_1_2_840_10008_2_16_10	

uid_1_2_840_10008_2_16_11	
uid_1_2_840_10008_2_16_12	
uid_1_2_840_10008_2_16_13	
uid_1_2_840_10008_2_16_14	
uid_1_2_840_10008_5_1_1_40	
uid_1_2_840_10008_5_1_1_40_1	
uid_1_2_840_10008_5_1_4_1_1_9_4_2	
uid_1_2_840_10008_5_1_4_1_1_9_5_1	
uid_1_2_840_10008_5_1_4_1_1_9_6_1	
uid_1_2_840_10008_5_1_4_1_1_11_5	
uid_1_2_840_10008_5_1_4_1_1_11_6	
uid_1_2_840_10008_1_2_4_104	
uid_1_2_840_10008_1_2_4_105	
uid_1_2_840_10008_1_2_4_106	
uid_1_2_840_10008_1_2_4_107	
uid_1_2_840_10008_1_2_4_108	
uid_1_2_840_10008_1_5_1	
uid_1_2_840_10008_5_1_4_1_1_11_7	
uid_1_2_840_10008_5_1_4_1_1_11_8	
uid_1_2_840_10008_5_1_4_1_1_11_9	
uid_1_2_840_10008_5_1_4_1_1_11_10	
uid_1_2_840_10008_5_1_4_1_1_11_11	
uid_1_2_840_10008_5_1_4_1_1_12_77	
uid_1_2_840_10008_5_1_4_1_1_13_1_4	
uid_1_2_840_10008_5_1_4_1_1_13_1_5	
uid_1_2_840_10008_5_1_4_1_1_14_1	
uid_1_2_840_10008_5_1_4_1_1_14_2	
uid_1_2_840_10008_5_1_4_1_1_30	
uid_1_2_840_10008_5_1_4_1_1_40	

uid_1_2_840_10008_5_1_4_1_1_66_6	
uid_1_2_840_10008_5_1_4_1_1_68_1	
uid_1_2_840_10008_5_1_4_1_1_68_2	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7	
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8	
uid_1_2_840_10008_5_1_4_1_1_78_1	
uid_1_2_840_10008_5_1_4_1_1_78_2	
uid_1_2_840_10008_5_1_4_1_1_78_3	
uid_1_2_840_10008_5_1_4_1_1_78_4	
uid_1_2_840_10008_5_1_4_1_1_78_5	
uid_1_2_840_10008_5_1_4_1_1_78_6	
uid_1_2_840_10008_5_1_4_1_1_78_7	
uid_1_2_840_10008_5_1_4_1_1_78_8	
uid_1_2_840_10008_5_1_4_1_1_79_1	
uid_1_2_840_10008_5_1_4_1_1_80_1	
uid_1_2_840_10008_5_1_4_1_1_81_1	
uid_1_2_840_10008_5_1_4_1_1_82_1	
uid_1_2_840_10008_5_1_4_1_1_88_34	
uid_1_2_840_10008_5_1_4_1_1_88_35	
uid_1_2_840_10008_5_1_4_1_1_88_68	
uid_1_2_840_10008_5_1_4_1_1_88_69	
uid_1_2_840_10008_5_1_4_1_1_88_70	
uid_1_2_840_10008_5_1_4_1_1_88_71	

uid_1_2_840_10008_5_1_4_1_1_88_72	
uid_1_2_840_10008_5_1_4_1_1_88_73	
uid_1_2_840_10008_5_1_4_1_1_88_74	
uid_1_2_840_10008_5_1_4_1_1_88_75	
uid_1_2_840_10008_5_1_4_1_1_90_1	
uid_1_2_840_10008_5_1_4_1_1_104_3	
uid_1_2_840_10008_5_1_4_1_1_130	
uid_1_2_840_10008_5_1_4_1_1_131	
uid_1_2_840_10008_5_1_4_1_1_200_1	
uid_1_2_840_10008_5_1_4_1_1_200_2	
uid_1_2_840_10008_5_1_4_1_1_200_3	
uid_1_2_840_10008_5_1_4_1_1_200_4	
uid_1_2_840_10008_5_1_4_1_1_200_5	
uid_1_2_840_10008_5_1_4_1_1_200_6	
uid_1_2_840_10008_5_1_4_1_1_481_10	
uid_1_2_840_10008_5_1_4_1_1_481_11	
uid_1_2_840_10008_5_1_4_1_1_501_1	
uid_1_2_840_10008_5_1_4_1_1_501_2_1	
uid_1_2_840_10008_5_1_4_1_1_501_2_2	
uid_1_2_840_10008_5_1_4_1_1_501_3	
uid_1_2_840_10008_5_1_4_1_1_501_4	
uid_1_2_840_10008_5_1_4_1_1_501_5	
uid_1_2_840_10008_5_1_4_1_1_501_6	
uid_1_2_840_10008_5_1_4_1_1_601_1	
uid_1_2_840_10008_5_1_4_1_1_601_2	
uid_1_2_840_10008_5_1_4_1_2_4_2	

uid_1_2_840_10008_5_1_4_1_2_4_3	
uid_1_2_840_10008_5_1_4_1_2_5_3	
uid_1_2_840_10008_5_1_4_20_1	
uid_1_2_840_10008_5_1_4_20_2	
uid_1_2_840_10008_5_1_4_20_3	
uid_1_2_840_10008_5_1_4_34_5_1	
uid_1_2_840_10008_5_1_4_34_6	
uid_1_2_840_10008_5_1_4_34_6_1	
uid_1_2_840_10008_5_1_4_34_6_2	
uid_1_2_840_10008_5_1_4_34_6_3	
uid_1_2_840_10008_5_1_4_34_6_4	
uid_1_2_840_10008_5_1_4_34_7	
uid_1_2_840_10008_5_1_4_34_8	
uid_1_2_840_10008_5_1_4_34_9	
uid_1_2_840_10008_5_1_4_34_10	
uid_1_2_840_10008_5_1_4_38_4	
uid_1_2_840_10008_5_1_4_39_1	
uid_1_2_840_10008_5_1_4_39_2	
uid_1_2_840_10008_5_1_4_39_3	
uid_1_2_840_10008_5_1_4_39_4	
uid_1_2_840_10008_5_1_4_43_1	
uid_1_2_840_10008_5_1_4_43_2	
uid_1_2_840_10008_5_1_4_43_3	

uid_1_2_840_10008_5_1_4_43_4	
uid_1_2_840_10008_5_1_4_44_1	
uid_1_2_840_10008_5_1_4_44_2	
uid_1_2_840_10008_5_1_4_44_3	
uid_1_2_840_10008_5_1_4_44_4	
uid_1_2_840_10008_5_1_4_45_1	
uid_1_2_840_10008_5_1_4_45_2	
uid_1_2_840_10008_5_1_4_45_3	
uid_1_2_840_10008_5_1_4_45_4	
uid_1_2_840_10008_7_1_1	
uid_1_2_840_10008_7_1_2	
uid_1_2_840_10008_8_1_1	
uid_1_2_840_10008_5_1_4_1_1_4_3	
uid_1_2_840_10008_15_1_1	

12.324.4 Constructor & Destructor Documentation

12.324.4.1 UIDs()

gdcmm::UIDs::UIDs () [default]

12.324.5 Member Function Documentation

12.324.5.1 GetName()

const char * gdcmm::UIDs::GetName () const

When object is Initialize function return the well known name associated with uid return NULL when not initialized

Examples

[GenerateStandardSOPClasses.cxx](#).

Referenced by [gdcmm::operator<<\(\)](#).

12.324.5.2 GetNumberOfTransferSyntaxStrings()

```
unsigned int gdcmm::UIDs::GetNumberOfTransferSyntaxStrings () [static]
```

12.324.5.3 GetString()

```
const char * gdcmm::UIDs::GetString () const
```

When object is Initialize function return the uid return NULL when not initialized

Examples

[GenerateStandardSOPClasses.cxx](#).

Referenced by [gdcmm::operator<<\(\)](#).

12.324.5.4 GetTransferSyntaxString()

```
const char *const * gdcmm::UIDs::GetTransferSyntaxString (  
    unsigned int ts) [static]
```

12.324.5.5 GetTransferSyntaxStrings()

```
TransferSyntaxStringsType gdcmm::UIDs::GetTransferSyntaxStrings () [static]
```

12.324.5.6 GetUIDName()

```
const char * gdcmm::UIDs::GetUIDName (  
    unsigned int ts) [static]
```

12.324.5.7 GetUIDString()

```
const char * gdcmm::UIDs::GetUIDString (  
    unsigned int ts) [static]
```

12.324.5.8 operator TSType()

```
gdcmm::UIDs::operator TSType () const [inline]
```

12.324.5.9 SetFromUID()

```
bool gdcm::UIDs::SetFromUID (  
    const char * str)
```

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

Examples

[GenerateStandardSOPClasses.cxx](#).

The documentation for this class was generated from the following file:

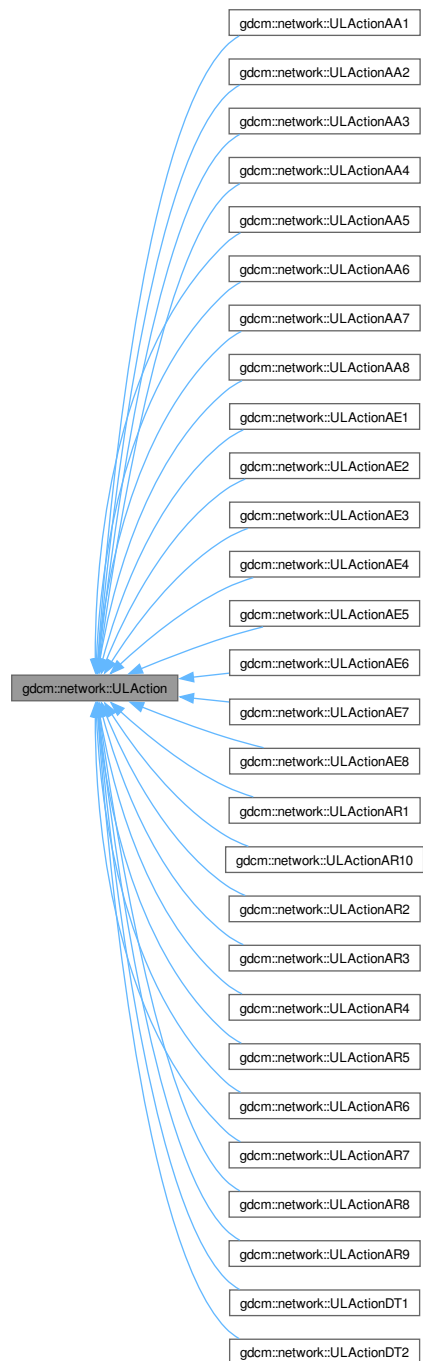
- [gdcmUIDs.h](#)

12.325 gdcm::network::ULAction Class Reference

[ULAction](#).

```
#include <gdcmULAction.h>
```

Inheritance diagram for `gdm::network::ULAction`:



Public Member Functions

- `ULAction()`=default

- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete
- virtual [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)=0

12.325.1 Detailed Description

[ULAction](#).

A [ULConnection](#) in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given [ULConnection](#).

Essentially, the [ULConnectionManager](#) will take this object, determined from the current ULState of the [ULConnection](#), and pass the [ULConnection](#) object to the [ULAction](#). The [ULAction](#) will then invoke whatever necessary commands are required by a given action.

The result of a [ULAction](#) is a [ULError](#) (ie, what happened as a result of the action).

This [ULError](#) is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads – be those filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcmm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

12.325.2 Constructor & Destructor Documentation

12.325.2.1 [ULAction](#)() [1/2]

`gdcmm::network::ULAction::ULAction ()` [default]

Referenced by [ULAction\(\)](#), and [operator=\(\)](#).

12.325.2.2 [~ULAction](#)()

`virtual gdcmm::network::ULAction::~~ULAction ()` [virtual], [default]

12.325.2.3 ULAction() [2/2]

```
gdcmm::network::ULAction::ULAction (
    const ULAction & inAction)    [delete]
```

References [ULAction\(\)](#).

12.325.3 Member Function Documentation

12.325.3.1 operator=()

```
void gdcmm::network::ULAction::operator= (
    const ULAction & )    [delete]
```

References [ULAction\(\)](#).

12.325.3.2 PerformAction()

```
virtual EStateID gdcmm::network::ULAction::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)    [pure virtual]
```

Implemented in [gdcmm::network::ULActionAA1](#), [gdcmm::network::ULActionAA2](#), [gdcmm::network::ULActionAA3](#), [gdcmm::network::ULActionAA4](#), [gdcmm::network::ULActionAA5](#), [gdcmm::network::ULActionAA6](#), [gdcmm::network::ULActionAA7](#), [gdcmm::network::ULActionAA8](#), [gdcmm::network::ULActionAE1](#), [gdcmm::network::ULActionAE2](#), [gdcmm::network::ULActionAE3](#), [gdcmm::network::ULActionAE4](#), [gdcmm::network::ULActionAE5](#), [gdcmm::network::ULActionAE6](#), [gdcmm::network::ULActionAE7](#), [gdcmm::network::ULActionAE8](#), [gdcmm::network::ULActionAR10](#), [gdcmm::network::ULActionAR1](#), [gdcmm::network::ULActionAR2](#), [gdcmm::network::ULActionAR3](#), [gdcmm::network::ULActionAR4](#), [gdcmm::network::ULActionAR5](#), [gdcmm::network::ULActionAR6](#), [gdcmm::network::ULActionAR7](#), [gdcmm::network::ULActionAR8](#), [gdcmm::network::ULActionAR9](#), [gdcmm::network::ULActionDT1](#) and [gdcmm::network::ULActionDT2](#).

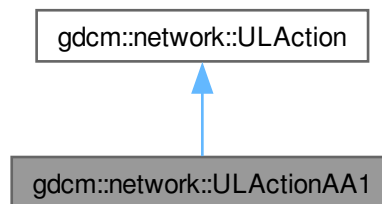
The documentation for this class was generated from the following file:

- [gdcmmULAction.h](#)

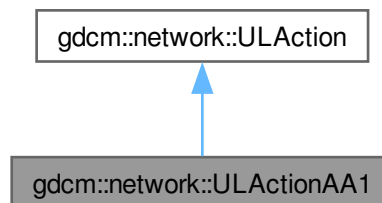
12.326 gdcm::network::ULActionAA1 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.326.1 Member Function Documentation

12.326.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA1::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

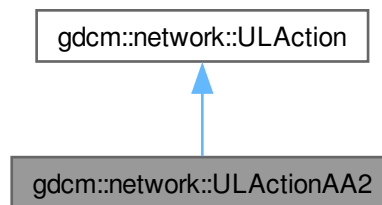
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

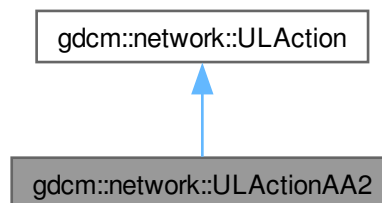
12.327 gdcmm::network::ULActionAA2 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA2:



Collaboration diagram for gdcmm::network::ULActionAA2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.327.1 Member Function Documentation

12.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

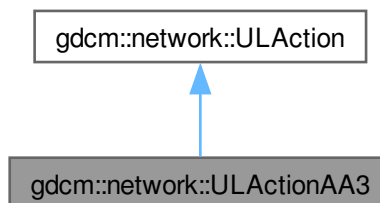
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

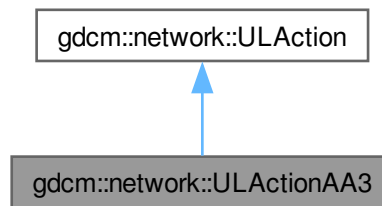
12.328 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for [gdcm::network::ULActionAA3](#):



Collaboration diagram for `gdcm::network::ULActionAA3`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.328.1 Member Function Documentation

12.328.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

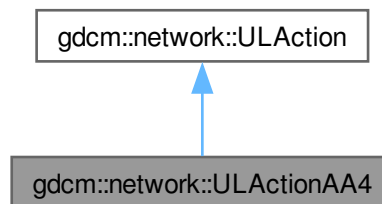
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

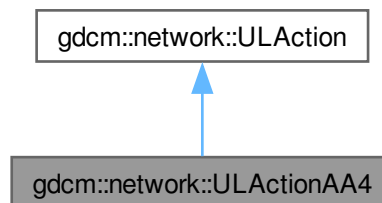
12.329 gdcn::network::ULActionAA4 Class Reference

```
#include <gdcnULActionAA.h>
```

Inheritance diagram for gdcn::network::ULActionAA4:



Collaboration diagram for gdcn::network::ULActionAA4:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.329.1 Member Function Documentation

12.329.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA4::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

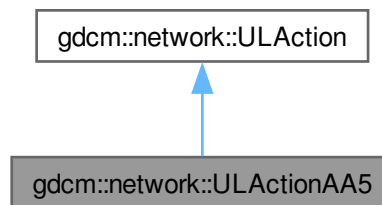
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

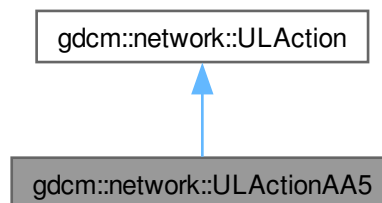
12.330 gdcmm::network::ULActionAA5 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA5:



Collaboration diagram for gdcmm::network::ULActionAA5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.330.1 Member Function Documentation

12.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

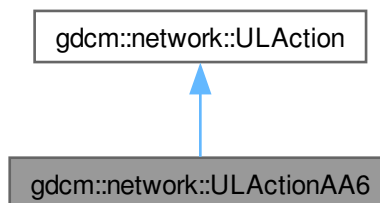
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

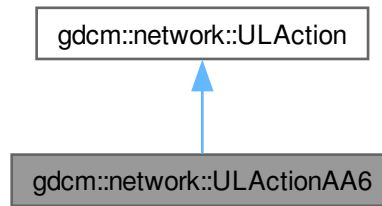
12.331 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for [gdcm::network::ULActionAA6](#):



Collaboration diagram for `gdcm::network::ULActionAA6`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.331.1 Member Function Documentation

12.331.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

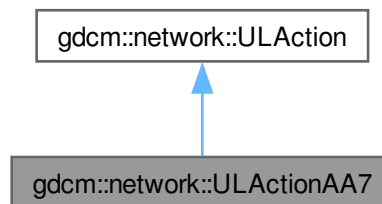
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

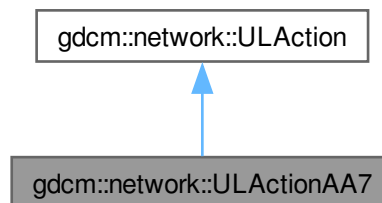
12.332 gdcm::network::ULActionAA7 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA7:



Collaboration diagram for gdcm::network::ULActionAA7:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.332.1 Member Function Documentation

12.332.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA7::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

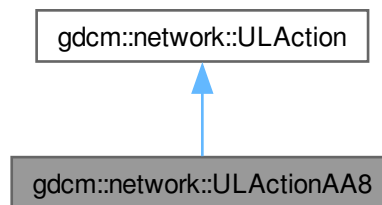
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

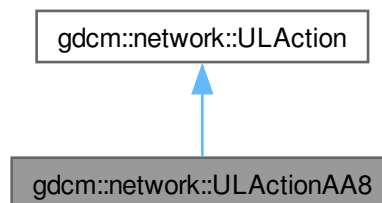
12.333 gdcmm::network::ULActionAA8 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA8:



Collaboration diagram for gdcmm::network::ULActionAA8:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.333.1 Member Function Documentation

12.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

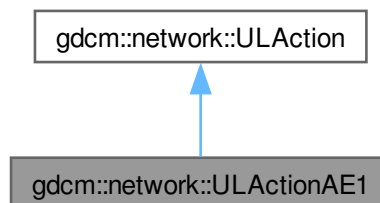
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

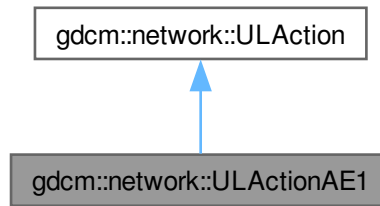
12.334 gdcm::network::ULActionAE1 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE1](#):



Collaboration diagram for `gdcm::network::ULActionAE1`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.334.1 Member Function Documentation

12.334.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

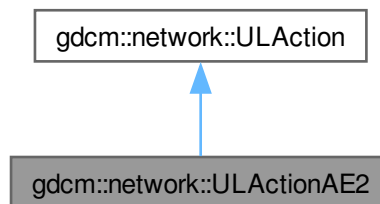
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

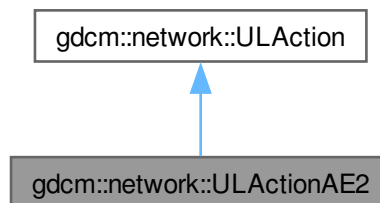
12.335 gdcn::network::ULActionAE2 Class Reference

```
#include <gdcnULActionAE.h>
```

Inheritance diagram for gdcn::network::ULActionAE2:



Collaboration diagram for gdcn::network::ULActionAE2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.335.1 Member Function Documentation

12.335.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE2::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

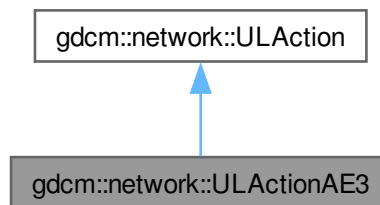
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

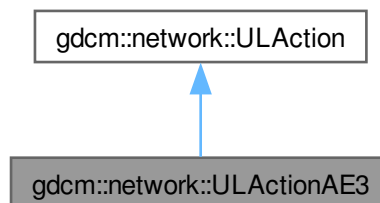
12.336 gdcmm::network::ULActionAE3 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE3:



Collaboration diagram for gdcmm::network::ULActionAE3:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.336.1 Member Function Documentation

12.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

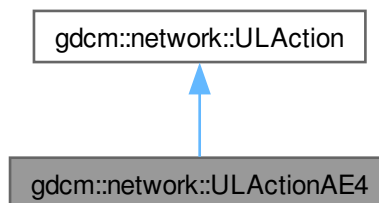
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

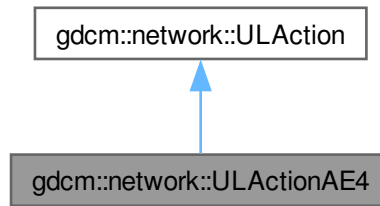
12.337 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE4](#):



Collaboration diagram for `gdcm::network::ULActionAE4`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.337.1 Member Function Documentation

12.337.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

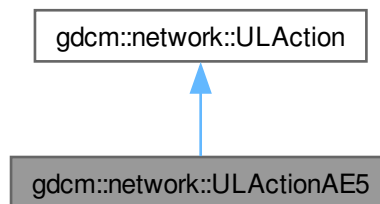
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

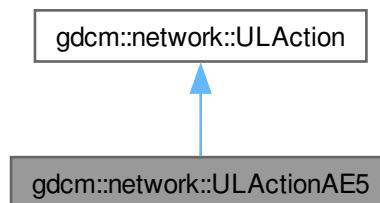
12.338 gdcm::network::ULActionAE5 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE5:



Collaboration diagram for gdcm::network::ULActionAE5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.338.1 Member Function Documentation

12.338.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE5::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

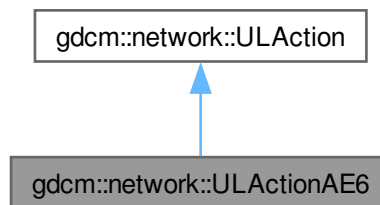
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

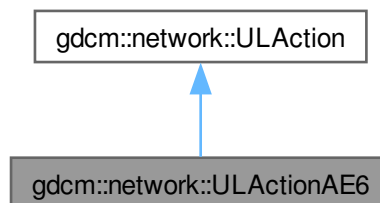
12.339 gdcmm::network::ULActionAE6 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for `gdcmm::network::ULActionAE6`:



Collaboration diagram for `gdcmm::network::ULActionAE6`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.339.1 Member Function Documentation

12.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

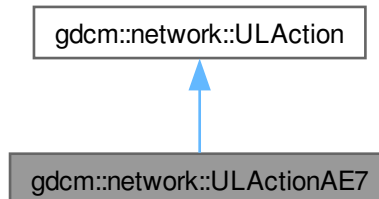
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

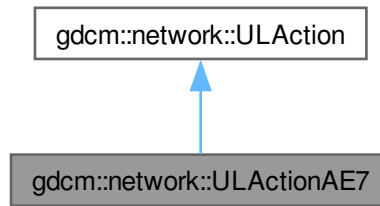
12.340 gdcm::network::ULActionAE7 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for [gdcm::network::ULActionAE7](#):



Collaboration diagram for `gdcm::network::ULActionAE7`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.340.1 Member Function Documentation

12.340.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

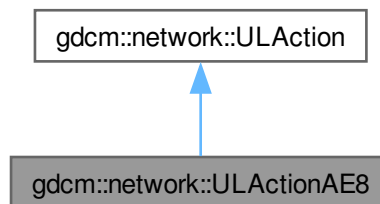
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

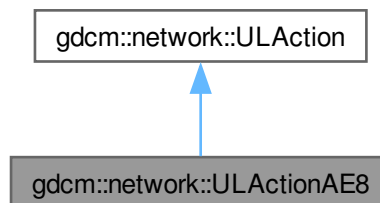
12.341 gdcn::network::ULActionAE8 Class Reference

```
#include <gdcnULActionAE.h>
```

Inheritance diagram for gdcn::network::ULActionAE8:



Collaboration diagram for gdcn::network::ULActionAE8:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.341.1 Member Function Documentation

12.341.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE8::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

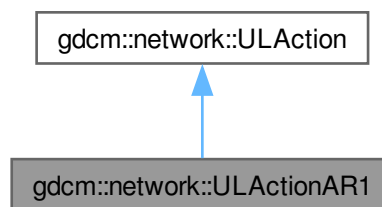
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

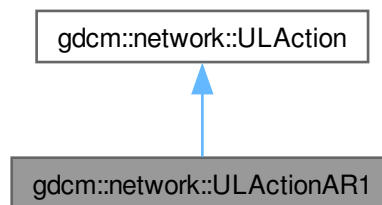
12.342 gdcmm::network::ULActionAR1 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR1:



Collaboration diagram for gdcmm::network::ULActionAR1:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.342.1 Member Function Documentation

12.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

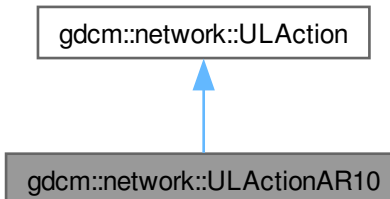
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

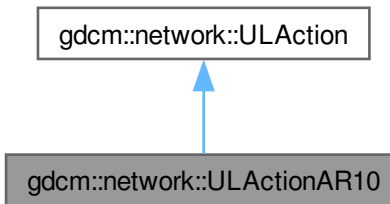
12.343 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR10](#):



Collaboration diagram for `gdcm::network::ULActionAR10`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.343.1 Member Function Documentation

12.343.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR10::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

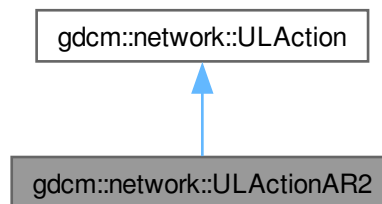
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

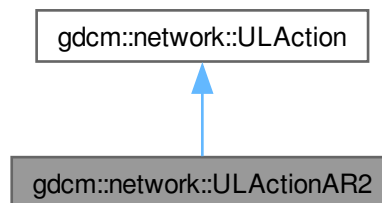
12.344 gdcn::network::ULActionAR2 Class Reference

```
#include <gdcnULActionAR.h>
```

Inheritance diagram for gdcn::network::ULActionAR2:



Collaboration diagram for gdcn::network::ULActionAR2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.344.1 Member Function Documentation

12.344.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR2::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

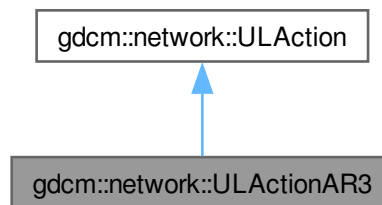
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

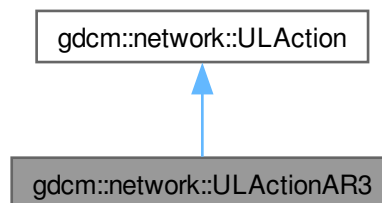
12.345 gdcmm::network::ULActionAR3 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR3:



Collaboration diagram for gdcmm::network::ULActionAR3:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.345.1 Member Function Documentation

12.345.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

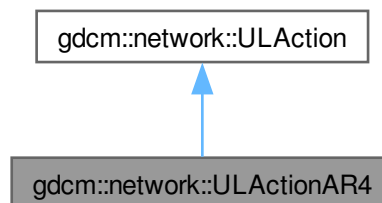
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

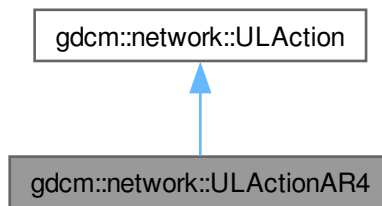
12.346 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR4](#):



Collaboration diagram for `gdcm::network::ULActionAR4`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.346.1 Member Function Documentation

12.346.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

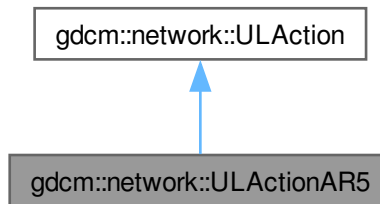
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

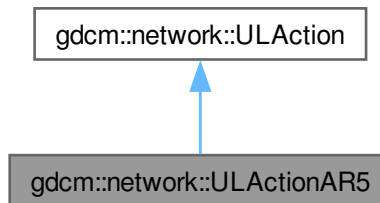
12.347 gdcn::network::ULActionAR5 Class Reference

```
#include <gdcnULActionAR.h>
```

Inheritance diagram for gdcn::network::ULActionAR5:



Collaboration diagram for gdcn::network::ULActionAR5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.347.1 Member Function Documentation

12.347.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR5::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

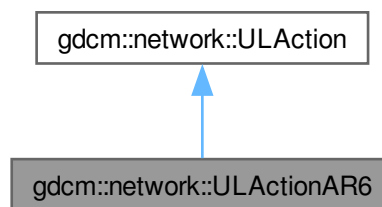
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

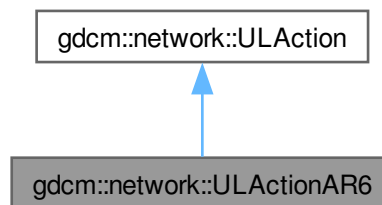
12.348 gdcmm::network::ULActionAR6 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR6:



Collaboration diagram for gdcmm::network::ULActionAR6:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.348.1 Member Function Documentation

12.348.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

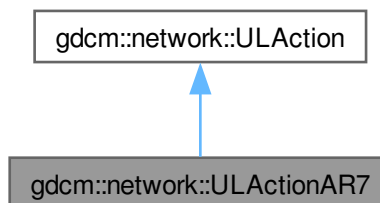
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

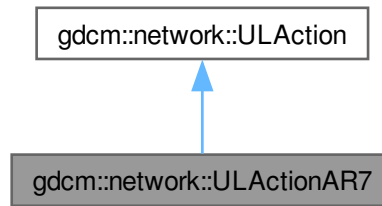
12.349 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for [gdcm::network::ULActionAR7](#):



Collaboration diagram for `gdcm::network::ULActionAR7`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.349.1 Member Function Documentation

12.349.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

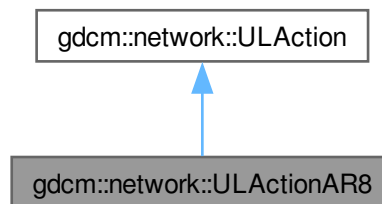
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

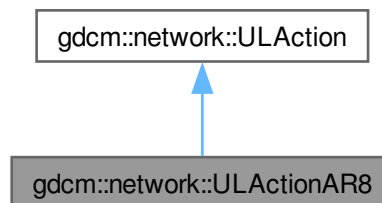
12.350 gdcn::network::ULActionAR8 Class Reference

```
#include <gdcnULActionAR.h>
```

Inheritance diagram for gdcn::network::ULActionAR8:



Collaboration diagram for gdcn::network::ULActionAR8:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.350.1 Member Function Documentation

12.350.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR8::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcmm::network::ULAction](#).

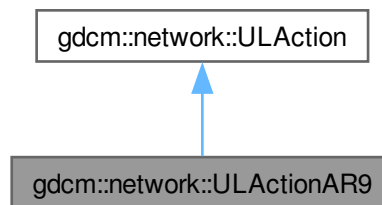
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

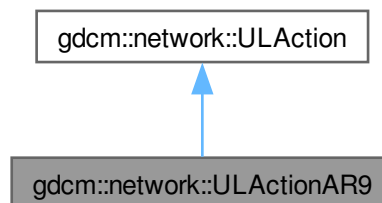
12.351 gdcmm::network::ULActionAR9 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR9:



Collaboration diagram for gdcmm::network::ULActionAR9:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.351.1 Member Function Documentation

12.351.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR9::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcm::network::ULAction](#).

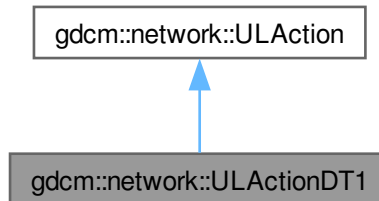
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

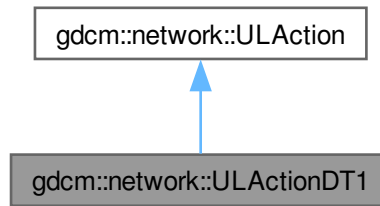
12.352 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for [gdcm::network::ULActionDT1](#):



Collaboration diagram for `gdcm::network::ULActionDT1`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcm::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.352.1 Member Function Documentation

12.352.1.1 PerformAction()

```

EStateID gdcm::network::ULActionDT1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) [override], [virtual]
  
```

Implements [gdcm::network::ULAction](#).

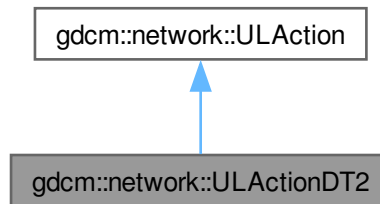
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

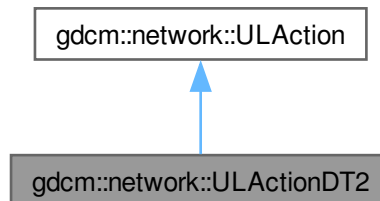
12.353 gdcn::network::ULActionDT2 Class Reference

```
#include <gdcnULActionDT.h>
```

Inheritance diagram for gdcn::network::ULActionDT2:



Collaboration diagram for gdcn::network::ULActionDT2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &out←WaitingForEvent, [EEventID](#) &outRaisedEvent) override

Public Member Functions inherited from [gdcn::network::ULAction](#)

- [ULAction](#) ()=default
- [ULAction](#) (const [ULAction](#) &inAction)=delete
- virtual [~ULAction](#) ()=default
- void [operator=](#) (const [ULAction](#) &)=delete

12.353.1 Member Function Documentation

12.353.1.1 PerformAction()

```
EStateID gdcn::network::ULActionDT2::PerformAction (  
    Subject * s,  
    ULEvent & inEvent,  
    ULConnection & inConnection,  
    bool & outWaitingForEvent,  
    EEventID & outRaisedEvent)  [override], [virtual]
```

Implements [gdcn::network::ULAction](#).

The documentation for this class was generated from the following file:

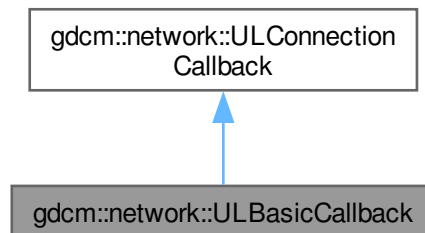
- [gdcnULActionDT.h](#)

12.354 gdcn::network::ULBasicCallback Class Reference

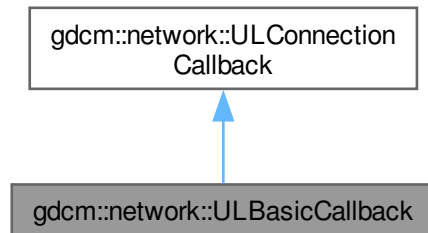
[ULBasicCallback](#).

```
#include <gdcnULBasicCallback.h>
```

Inheritance diagram for gdcn::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- [ULBasicCallback](#) ()=default
- [~ULBasicCallback](#) () override=default
- `std::vector< DataSet > const & GetDataSets () const`
- `std::vector< DataSet > const & GetResponses () const`
- `void HandleDataSet (const DataSet &inDataSet) override`
- `void HandleResponse (const DataSet &inDataSet) override`

Public Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- [ULConnectionCallback](#) ()
- `virtual ~ULConnectionCallback ()=default`
- `bool DataSetHandles () const`
- `void ResetHandledDataSet ()`
- `void SetImplicitFlag (const bool imp)`

Additional Inherited Members

Protected Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- `void DataSetHandled ()`

Protected Attributes inherited from [gdcm::network::ULConnectionCallback](#)

- `bool mImplicit`

12.354.1 Detailed Description

[ULBasicCallback](#).

This is the most basic of callbacks for how the [ULConnectionManager](#) handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the [ULConnectionManager](#).

12.354.2 Constructor & Destructor Documentation

12.354.2.1 [ULBasicCallback\(\)](#)

gdcm::network::ULBasicCallback::ULBasicCallback () [default]

12.354.2.2 [~ULBasicCallback\(\)](#)

gdcm::network::ULBasicCallback::~~ULBasicCallback () [override], [default]

12.354.3 Member Function Documentation

12.354.3.1 [GetDataSets\(\)](#)

std::vector< [DataSet](#) > const & gdcm::network::ULBasicCallback::GetDataSets () const

12.354.3.2 [GetResponses\(\)](#)

std::vector< [DataSet](#) > const & gdcm::network::ULBasicCallback::GetResponses () const

12.354.3.3 [HandleDataSet\(\)](#)

void gdcm::network::ULBasicCallback::HandleDataSet (
const [DataSet](#) & inDataSet) [override], [virtual]

Implements [gdcm::network::ULConnectionCallback](#).

12.354.3.4 [HandleResponse\(\)](#)

void gdcm::network::ULBasicCallback::HandleResponse (
const [DataSet](#) & inDataSet) [override], [virtual]

Implements [gdcm::network::ULConnectionCallback](#).

The documentation for this class was generated from the following file:

- [gdcmULBasicCallback.h](#)

12.355 gdcm::network::ULConnection Class Reference

[ULConnection](#).

```
#include <gdcmULConnection.h>
```

Public Member Functions

- [ULConnection](#) (const [ULConnection](#) &)=delete
- [ULConnection](#) (const [ULConnectionInfo](#) &inUserInformation)
- virtual [~ULConnection](#) ()
- void [AddAcceptedPresentationContext](#) (const [PresentationContextAC](#) &inPC)
- [PresentationContextRQ](#) [FindContext](#) (const [DataElement](#) &de) const
- std::vector< [PresentationContextAC](#) > & [GetAcceptedPresentationContexts](#) ()
- std::vector< [PresentationContextAC](#) > const & [GetAcceptedPresentationContexts](#) () const
- const [ULConnectionInfo](#) & [GetConnectionInfo](#) () const
- uint32_t [GetMaxPDUSize](#) () const
- const [PresentationContextAC](#) * [GetPresentationContextACByID](#) (uint8_t id) const
- uint8_t [GetPresentationContextIDFromPresentationContext](#) ([PresentationContextRQ](#) const &pc) const
- return 0 upon error
- const [PresentationContextRQ](#) * [GetPresentationContextRQByID](#) (uint8_t id) const
- std::vector< [PresentationContextRQ](#) > const & [GetPresentationContexts](#) () const
- std::iostream * [GetProtocol](#) ()
- [EStateID](#) [GetState](#) () const
- [ARTIMTimer](#) & [GetTimer](#) ()
- bool [InitializeConnection](#) ()
- used to establish scu connections
- bool [InitializeIncomingConnection](#) ()
- used to establish scp connections
- void [operator=](#) (const [ULConnection](#) &)=delete
- void [SetMaxPDUSize](#) (uint32_t inSize)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContext](#) > &inContexts)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContextRQ](#) > &inContexts)
- void [SetState](#) (const [EStateID](#) &inState)
- void [StopProtocol](#) ()

Friends

- class [ULActionAE6](#)
- class [ULConnectionManager](#)

12.355.1 Detailed Description

[ULConnection](#).

This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The [ULConnectionManager](#) tells the [ULConnection](#) what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a [ULSecureConnection](#), if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a [gdcm](#) object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future [ULSecureConnection](#) warrants the addition, rather than having everything be managed from within the [ULConnectionManager](#) (or this class) without a wrapper.

12.355.2 Constructor & Destructor Documentation

12.355.2.1 [ULConnection\(\)](#) [1/2]

```
gdcm::network::ULConnection::ULConnection (
    const ULConnectionInfo & inUserInformation)
```

Referenced by [ULConnection\(\)](#), and [operator=\(\)](#).

12.355.2.2 [~ULConnection\(\)](#)

```
virtual gdcm::network::ULConnection::~~ULConnection () [virtual]
```

12.355.2.3 [ULConnection\(\)](#) [2/2]

```
gdcm::network::ULConnection::ULConnection (
    const ULConnection & ) [delete]
```

References [ULConnection\(\)](#).

12.355.3 Member Function Documentation

12.355.3.1 [AddAcceptedPresentationContext\(\)](#)

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
    const PresentationContextAC & inPC)
```

12.355.3.2 FindContext()

[PresentationContextRQ](#) gdcm::network::ULConnection::FindContext (
const [DataElement](#) & de) const

12.355.3.3 GetAcceptedPresentationContexts() [1/2]

std::vector< [PresentationContextAC](#) > & gdcm::network::ULConnection::GetAcceptedPresentationContexts ()

12.355.3.4 GetAcceptedPresentationContexts() [2/2]

std::vector< [PresentationContextAC](#) > const & gdcm::network::ULConnection::GetAcceptedPresentationContexts () const

12.355.3.5 GetConnectionInfo()

const [ULConnectionInfo](#) & gdcm::network::ULConnection::GetConnectionInfo () const

12.355.3.6 GetMaxPDUSize()

uint32_t gdcm::network::ULConnection::GetMaxPDUSize () const

12.355.3.7 GetPresentationContextACByID()

const [PresentationContextAC](#) * gdcm::network::ULConnection::GetPresentationContextACByID (
uint8_t id) const

12.355.3.8 GetPresentationContextIDFromPresentationContext()

uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
[PresentationContextRQ](#) const & pc) const

return 0 upon error

12.355.3.9 GetPresentationContextRQByID()

const [PresentationContextRQ](#) * gdcm::network::ULConnection::GetPresentationContextRQByID (
uint8_t id) const

12.355.3.10 GetPresentationContexts()

std::vector< [PresentationContextRQ](#) > const & gdcm::network::ULConnection::GetPresentationContexts () const

12.355.3.11 GetProtocol()

`std::iostream * gdcm::network::ULConnection::GetProtocol ()`

12.355.3.12 GetState()

`EStateID gdcm::network::ULConnection::GetState () const`

12.355.3.13 GetTimer()

`ARTIMTimer & gdcm::network::ULConnection::GetTimer ()`

12.355.3.14 InitializeConnection()

`bool gdcm::network::ULConnection::InitializeConnection ()`

used to establish scu connections

12.355.3.15 InitializeIncomingConnection()

`bool gdcm::network::ULConnection::InitializeIncomingConnection ()`

used to establish scp connections

12.355.3.16 operator=()

`void gdcm::network::ULConnection::operator= (
 const ULConnection &) [delete]`

References [ULConnection\(\)](#).

12.355.3.17 SetMaxPDUSize()

`void gdcm::network::ULConnection::SetMaxPDUSize (
 uint32_t inSize)`

12.355.3.18 SetPresentationContexts() [1/2]

`void gdcm::network::ULConnection::SetPresentationContexts (
 const std::vector< PresentationContext > & inContexts)`

12.355.3.19 SetPresentationContexts() [2/2]

```
void gdcmm::network::ULConnection::SetPresentationContexts (  
    const std::vector< PresentationContextRQ > & inContexts)
```

12.355.3.20 SetState()

```
void gdcmm::network::ULConnection::SetState (  
    const EStateID & inState)
```

12.355.3.21 StopProtocol()

```
void gdcmm::network::ULConnection::StopProtocol ()
```

12.355.4 Friends And Related Symbol Documentation

12.355.4.1 ULActionAE6

friend class ULActionAE6 [friend]

References [ULActionAE6](#).

Referenced by [ULActionAE6](#).

12.355.4.2 ULConnectionManager

friend class ULConnectionManager [friend]

References [ULConnectionManager](#).

Referenced by [ULConnectionManager](#).

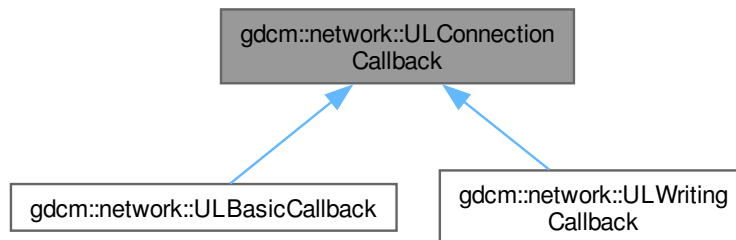
The documentation for this class was generated from the following file:

- [gdcmmULConnection.h](#)

12.356 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()=default
- bool [DataSetHandles](#) () const
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)=0
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)=0
- void [ResetHandledDataSet](#) ()
- void [SetImplicitFlag](#) (const bool imp)

Protected Member Functions

- void [DataSetHandled](#) ()

Protected Attributes

- bool [mImplicit](#)

12.356.1 Detailed Description

When a dataset comes back from a query/move/etc, the result can either be stored entirely in memory, or could be stored on disk. This class provides a mechanism to indicate what the [ULConnectionManager](#) should do with datasets that are produced through query results. The [ULConnectionManager](#) will call the [HandleDataSet](#) function during the course of receiving datasets. Particular implementations should fill in what that function does, including updating progress, etc. NOTE: since cmove requires that multiple event loops be employed, the callback function MUST set [mHandledDataSet](#) to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

12.356.2 Constructor & Destructor Documentation

12.356.2.1 ULConnectionCallback()

gdcm::network::ULConnectionCallback::ULConnectionCallback () [inline]

References [mImplicit](#).

12.356.2.2 ~ULConnectionCallback()

virtual gdcm::network::ULConnectionCallback::~~ULConnectionCallback () [virtual], [default]

12.356.3 Member Function Documentation

12.356.3.1 DataSetHandled()

void gdcm::network::ULConnectionCallback::DataSetHandled () [inline], [protected]

12.356.3.2 DataSetHandles()

bool gdcm::network::ULConnectionCallback::DataSetHandles () const [inline]

12.356.3.3 HandleDataSet()

virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
const [DataSet](#) & inDataSet) [pure virtual]

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

12.356.3.4 HandleResponse()

virtual void gdcm::network::ULConnectionCallback::HandleResponse (
const [DataSet](#) & inDataSet) [pure virtual]

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

12.356.3.5 ResetHandledDataSet()

void gdcm::network::ULConnectionCallback::ResetHandledDataSet () [inline]

12.356.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
    const bool imp) [inline]
```

References [mImplicit](#).

12.356.4 Member Data Documentation

12.356.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit [protected]
```

Referenced by [ULConnectionCallback\(\)](#), and [SetImplicitFlag\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmULConnectionCallback.h](#)

12.357 gdcm::network::ULConnectionInfo Class Reference

[ULConnectionInfo](#).

```
#include <gdcmULConnectionInfo.h>
```

Public Member Functions

- [ULConnectionInfo](#) ()
- const char * [GetCalledAETitle](#) () const
- std::string [GetCalledComputerName](#) () const
- unsigned long [GetCalledIPAddress](#) () const
- int [GetCalledIPPort](#) () const
- const char * [GetCallingAETitle](#) () const
- unsigned long [GetMaxPDULength](#) () const
- bool [Initialize](#) ([UserInformation](#) const &inUserInformation, const char *inCalledAETitle, const char *inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

12.357.1 Detailed Description

[ULConnectionInfo](#).

this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

12.357.2 Constructor & Destructor Documentation

12.357.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ()
```

12.357.3 Member Function Documentation

12.357.3.1 GetCalledAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCalledAETitle () const
```

12.357.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName () const
```

12.357.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress () const
```

12.357.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort () const
```

12.357.3.5 GetCallingAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCallingAETitle () const
```

12.357.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength () const
```

12.357.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (  
    UserInformation const & inUserInformation,  
    const char * inCalledAETitle,  
    const char * inCallingAETitle,  
    unsigned long inCalledIPAddress,  
    int inCalledIPPort,  
    std::string inCalledComputerName)
```

12.357.3.8 SetMaxPDULength()

```
void gdcmm::network::ULConnectionInfo::SetMaxPDULength (  
    unsigned long inMaxPDULength)
```

The documentation for this class was generated from the following file:

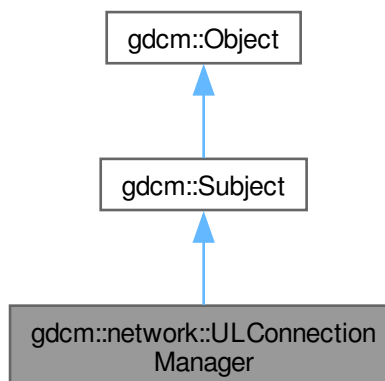
- [gdcmmULConnectionInfo.h](#)

12.358 gdcmm::network::ULConnectionManager Class Reference

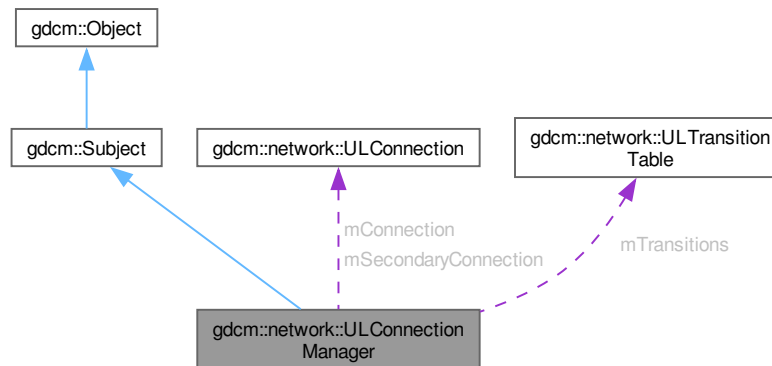
[ULConnectionManager](#).

```
#include <gdcmmULConnectionManager.h>
```

Inheritance diagram for gdcmm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- [ULConnectionManager](#) ()
- [~ULConnectionManager](#) () override
- bool [BreakConnection](#) (const double &inTimeout)
- void [BreakConnectionNow](#) ()
- bool [EstablishConnection](#) (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< [PresentationContext](#) > const &pcVector)
- bool [EstablishConnectionMove](#) (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, uint16_t inReturnPort, std::vector< [PresentationContext](#) > const &pcVector)
- std::vector< [PresentationDataValue](#) > [SendEcho](#) ()
- std::vector< [DataSet](#) > [SendFind](#) (const [BaseRootQuery](#) *inRootQuery)
- void [SendFind](#) (const [BaseRootQuery](#) *inRootQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendMove](#) (const [BaseRootQuery](#) *inRootQuery)
- bool [SendMove](#) (const [BaseRootQuery](#) *inRootQuery, [ULConnectionCallback](#) *inCallback)
- return false upon error
- std::vector< [DataSet](#) > [SendNAction](#) (const [BaseQuery](#) *inQuery)
- void [SendNAction](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendNCreate](#) (const [BaseQuery](#) *inQuery)
- void [SendNCreate](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendNDelete](#) (const [BaseQuery](#) *inQuery)
- void [SendNDelete](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendNEventReport](#) (const [BaseQuery](#) *inQuery)
- void [SendNEventReport](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendNGet](#) (const [BaseQuery](#) *inQuery)
- void [SendNGet](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)
- std::vector< [DataSet](#) > [SendNSet](#) (const [BaseQuery](#) *inQuery)
- void [SendNSet](#) (const [BaseQuery](#) *inQuery, [ULConnectionCallback](#) *inCallback)

- `std::vector< DataSet > SendStore` (const `File` &file, `std::istream *pStream=NULLPTR`, `std::streampos dataSetOffset=0`)
- `void SendStore` (const `File` &file, `ULConnectionCallback *inCallback`, `std::istream *pStream=NULLPTR`, `std::streampos dataSetOffset=0`)
callback based API

Public Member Functions inherited from `gdcM::Subject`

- `Subject` ()
- `~Subject` () override
- unsigned long `AddObserver` (const `Event` &event, `Command *`)
- unsigned long `AddObserver` (const `Event` &event, `Command *`) const
- `Command *` `GetCommand` (unsigned long tag)
- bool `HasObserver` (const `Event` &event) const
- void `InvokeEvent` (const `Event` &)
- void `InvokeEvent` (const `Event` &) const
- void `RemoveAllObservers` ()
- void `RemoveObserver` (unsigned long tag)

Public Member Functions inherited from `gdcM::Object`

- `Object` ()
- `Object` (const `Object` &)
Special requirement for copy/cstor, assignment operator.
- virtual `~Object` ()
- void `operator=` (const `Object` &)
- virtual void `Print` (`std::ostream` &) const

Protected Member Functions

- `ULConnectionManager` (const `ULConnectionManager` &inCM)
- `EStateID RunEventLoop` (`ULEvent` &inEvent, `ULConnection` *inWhichConnection, `ULConnectionCallback` *inCallback, const bool &startWaiting)
- `EStateID RunMoveEventLoop` (`ULEvent` &inEvent, `ULConnectionCallback` *inCallback)

Protected Member Functions inherited from `gdcM::Object`

- void `Register` ()
- void `UnRegister` ()

Protected Attributes

- `ULConnection *` `mConnection`
- `ULConnection *` `mSecondaryConnection`
- `ULTransitionTable` `mTransitions`

12.358.1 Detailed Description

[ULConnectionManager](#).

The [ULConnectionManager](#) performs actions on the [ULConnection](#) given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Its inputs are ULEvents, and it performs ULActions.

12.358.2 Constructor & Destructor Documentation

12.358.2.1 [ULConnectionManager\(\)](#) [1/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager (  
    const ULConnectionManager & inCM)  [protected]
```

References [ULConnectionManager\(\)](#).

Referenced by [ULConnectionManager\(\)](#).

12.358.2.2 [ULConnectionManager\(\)](#) [2/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager ()
```

12.358.2.3 [~ULConnectionManager\(\)](#)

```
gdcm::network::ULConnectionManager::~~ULConnectionManager ()  [override]
```

12.358.3 Member Function Documentation

12.358.3.1 [BreakConnection\(\)](#)

```
bool gdcm::network::ULConnectionManager::BreakConnection (  
    const double & inTimeout)
```

12.358.3.2 [BreakConnectionNow\(\)](#)

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ()
```

12.358.3.3 EstablishConnection()

```
bool gdcn::network::ULConnectionManager::EstablishConnection (
    const std::string & inAETitle,
    const std::string & inConnectAETitle,
    const std::string & inComputerName,
    long inIPAddress,
    uint16_t inConnectPort,
    double inTimeout,
    std::vector< PresentationContext > const & pcVector)
```

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'— have to give a return port for move to work as specified.

12.358.3.4 EstablishConnectionMove()

```
bool gdcn::network::ULConnectionManager::EstablishConnectionMove (
    const std::string & inAETitle,
    const std::string & inConnectAETitle,
    const std::string & inComputerName,
    long inIPAddress,
    uint16_t inConnectPort,
    double inTimeout,
    uint16_t inReturnPort,
    std::vector< PresentationContext > const & pcVector)
```

returns true for above reasons, but contains the special 'move' port

12.358.3.5 RunEventLoop()

```
EStateID gdcn::network::ULConnectionManager::RunEventLoop (
    ULEvent & inEvent,
    ULConnection * inWhichConnection,
    ULConnectionCallback * inCallback,
    const bool & startWaiting) [protected]
```

12.358.3.6 RunMoveEventLoop()

```
EStateID gdcn::network::ULConnectionManager::RunMoveEventLoop (
    ULEvent & inEvent,
    ULConnectionCallback * inCallback) [protected]
```

12.358.3.7 SendEcho()

```
std::vector< PresentationDataValue > gdcn::network::ULConnectionManager::SendEcho ()
```


12.358.3.8 SendFind() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendFind (  
    const BaseRootQuery * inRootQuery)
```

12.358.3.9 SendFind() [2/2]

```
void gdcmm::network::ULConnectionManager::SendFind (  
    const BaseRootQuery * inRootQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.10 SendMove() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendMove (  
    const BaseRootQuery * inRootQuery)
```

12.358.3.11 SendMove() [2/2]

```
bool gdcmm::network::ULConnectionManager::SendMove (  
    const BaseRootQuery * inRootQuery,  
    ULConnectionCallback * inCallback)
```

return false upon error

12.358.3.12 SendNAction() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendNAction (  
    const BaseQuery * inQuery)
```

12.358.3.13 SendNAction() [2/2]

```
void gdcmm::network::ULConnectionManager::SendNAction (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.14 SendNCreate() [1/2]

```
std::vector< DataSet > gdcmm::network::ULConnectionManager::SendNCreate (  
    const BaseQuery * inQuery)
```

12.358.3.15 SendNCreate() [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.16 SendNDelete() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNDelete (  
    const BaseQuery * inQuery)
```

12.358.3.17 SendNDelete() [2/2]

```
void gdcm::network::ULConnectionManager::SendNDelete (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.18 SendNEventReport() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNEventReport (  
    const BaseQuery * inQuery)
```

12.358.3.19 SendNEventReport() [2/2]

```
void gdcm::network::ULConnectionManager::SendNEventReport (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.20 SendNGet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNGet (  
    const BaseQuery * inQuery)
```

12.358.3.21 SendNGet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.22 SendNSet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNSet (  
    const BaseQuery * inQuery)
```

12.358.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (  
    const BaseQuery * inQuery,  
    ULConnectionCallback * inCallback)
```

12.358.3.24 SendStore() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendStore (  
    const File & file,  
    std::istream * pStream = nullptr,  
    std::streampos dataSetOffset = 0)
```

12.358.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (  
    const File & file,  
    ULConnectionCallback * inCallback,  
    std::istream * pStream = nullptr,  
    std::streampos dataSetOffset = 0)
```

callback based API

12.358.4 Member Data Documentation

12.358.4.1 mConnection

[ULConnection](#)* gdcm::network::ULConnectionManager::mConnection [protected]

12.358.4.2 mSecondaryConnection

[ULConnection](#)* gdcm::network::ULConnectionManager::mSecondaryConnection [protected]

12.358.4.3 mTransitions

[ULTransitionTable](#) gdcm::network::ULConnectionManager::mTransitions [protected]

The documentation for this class was generated from the following file:

- [gdcmULConnectionManager.h](#)

12.359 gdcm::network::ULEvent Class Reference

[ULEvent](#).

```
#include <gdcmULEvent.h>
```

Public Member Functions

- [ULEvent](#) (const [EEventID](#) &inEventID, [BasePDU](#) *inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- [ULEvent](#) (const [EEventID](#) &inEventID, std::vector< [BasePDU](#) * > inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- [~ULEvent](#) ()
- std::streampos [GetDataSetPos](#) () const
- [EEventID](#) [GetEvent](#) () const
- std::istream * [GetIStream](#) () const
- std::vector< [BasePDU](#) * > const & [GetPDUs](#) () const
- void [SetEvent](#) (const [EEventID](#) &inEvent)
- void [SetPDU](#) (std::vector< [BasePDU](#) * > const &inPDU)

12.359.1 Detailed Description

[ULEvent](#).

base class for network events.

An event consists of the event ID and the data associated with that event.

Note that once a PDU is created, it is now the responsibility of the associated event to destroy it!

12.359.2 Constructor & Destructor Documentation

12.359.2.1 ULEvent() [1/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    std::vector< BasePDU * > inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0) [inline]
```

12.359.2.2 ULEvent() [2/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    BasePDU * inBasePDU,
    std::istream * iStream = nullptr,
    std::streampos posDataSet = 0) [inline]
```

12.359.2.3 ~ULEvent()

gdcm::network::ULEvent::~~ULEvent () [inline]

12.359.3 Member Function Documentation

12.359.3.1 GetDataSetPos()

std::streampos gdcm::network::ULEvent::GetDataSetPos () const [inline]

12.359.3.2 GetEvent()

[EEventID](#) gdcm::network::ULEvent::GetEvent () const [inline]

12.359.3.3 GetIStream()

std::istream * gdcm::network::ULEvent::GetIStream () const [inline]

12.359.3.4 GetPDUs()

std::vector< [BasePDU](#) * > const & gdcm::network::ULEvent::GetPDUs () const [inline]

12.359.3.5 SetEvent()

void gdcm::network::ULEvent::SetEvent (
const [EEventID](#) & inEvent) [inline]

12.359.3.6 SetPDU()

void gdcm::network::ULEvent::SetPDU (
std::vector< [BasePDU](#) * > const & inPDU) [inline]

The documentation for this class was generated from the following file:

- [gdcmULEvent.h](#)

12.360 gdcm::network::ULTransitionTable Class Reference

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmULTransitionTable.h>
```

Public Member Functions

- [ULTransitionTable](#) ()
- void [HandleEvent](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

12.360.1 Detailed Description

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

Based roughly on the solutions in player2.cpp in the boost examples and this so question: <http://stackoverflow.com/questions/1647631/c-state-machine-design>

The transition table is constructed of TableRows. Each row is based on an event, and an event handler in the TransitionTable object takes a given event, and then finds the given row.

Then, given the current state of the connection, determines the appropriate action to take and then the state to transition to next.

12.360.2 Constructor & Destructor Documentation

12.360.2.1 ULTransitionTable()

```
gdcmm::network::ULTransitionTable::ULTransitionTable ()
```

12.360.3 Member Function Documentation

12.360.3.1 HandleEvent()

```
void gdcmm::network::ULTransitionTable::HandleEvent (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent) const
```

12.360.3.2 PrintTable()

```
void gdcmm::network::ULTransitionTable::PrintTable () const
```

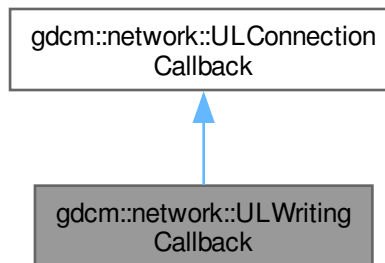
The documentation for this class was generated from the following file:

- [gdcmmULTransitionTable.h](#)

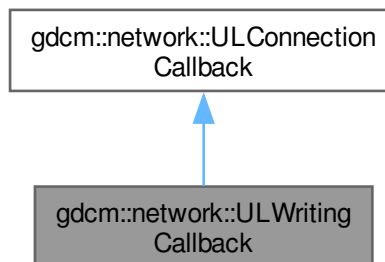
12.361 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- [ULWritingCallback](#) ()=default
- [~ULWritingCallback](#) () override=default
- void [HandleDataSet](#) (const [DataSet](#) &inDataSet) override
- void [HandleResponse](#) (const [DataSet](#) &inDataSet) override
- void [SetDirectory](#) (const std::string &inDirectoryName)

provide the directory into which all files are written.

Public Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()=default
- bool [DataSetHandles](#) () const
- void [ResetHandledDataSet](#) ()
- void [SetImplicitFlag](#) (const bool imp)

Additional Inherited Members

Protected Member Functions inherited from [gdcm::network::ULConnectionCallback](#)

- void [DataSetHandled](#) ()

Protected Attributes inherited from [gdcm::network::ULConnectionCallback](#)

- bool [mImplicit](#)

12.361.1 Constructor & Destructor Documentation

12.361.1.1 [ULWritingCallback](#)()

[gdcm::network::ULWritingCallback::ULWritingCallback](#) () [default]

12.361.1.2 [~ULWritingCallback](#)()

[gdcm::network::ULWritingCallback::~~ULWritingCallback](#) () [override], [default]

12.361.2 Member Function Documentation

12.361.2.1 [HandleDataSet](#)()

void [gdcm::network::ULWritingCallback::HandleDataSet](#) (
const [DataSet](#) & inDataSet) [override], [virtual]

Implements [gdcm::network::ULConnectionCallback](#).

12.361.2.2 [HandleResponse](#)()

void [gdcm::network::ULWritingCallback::HandleResponse](#) (
const [DataSet](#) & inDataSet) [override], [virtual]

Implements [gdcm::network::ULConnectionCallback](#).

12.361.2.3 SetDirectory()

```
void gdcm::network::ULWritingCallback::SetDirectory (  
    const std::string & inDirectoryName) [inline]
```

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

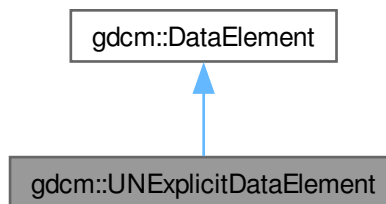
- [gdcmULWritingCallback.h](#)

12.362 gdcm::UNExplicitDataElement Class Reference

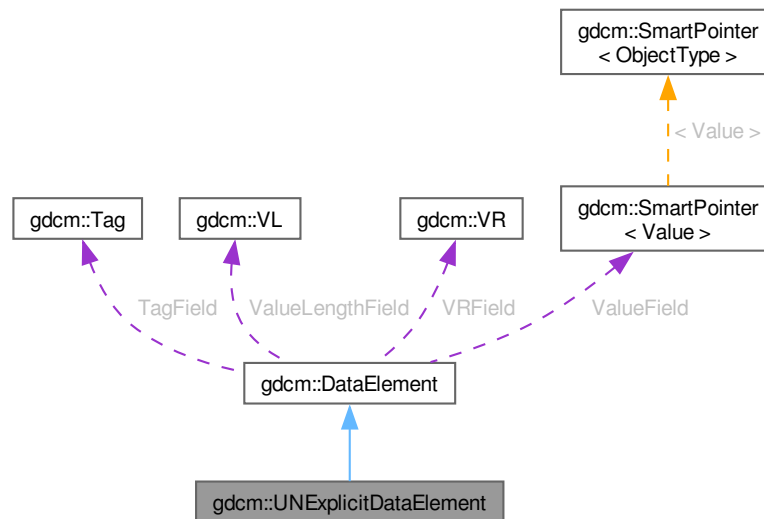
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for `gdcm::UNExplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const

- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.362.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

Note

bla

12.362.2 Member Function Documentation

12.362.2.1 GetLength()

[VL](#) [gdcm::UNExplicitDataElement::GetLength](#) () const

12.362.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::Read (
    std::istream & is)
```

12.362.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.362.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```

12.362.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length)
```

The documentation for this class was generated from the following file:

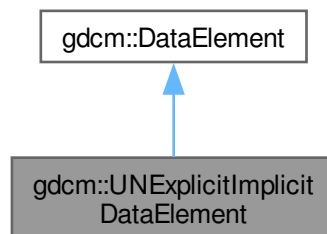
- [gdcmUNExplicitDataElement.h](#)

12.363 gdcm::UNExplicitImplicitDataElement Class Reference

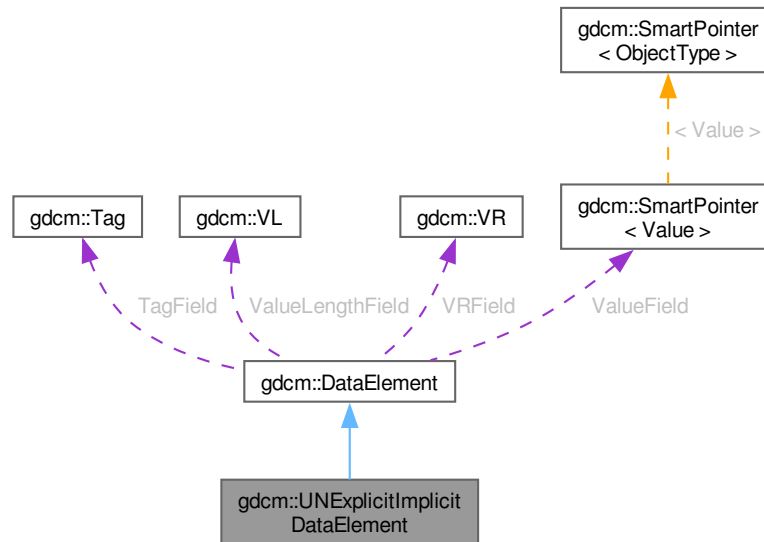
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitImplicitDataElement:



Collaboration diagram for `gdcm::UNExplicitImplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is)

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const DataElement &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make Data [Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const
- [Tag](#) & [GetTag](#) ()

- const [Tag](#) & [GetTag](#) () const
Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.363.1 Detailed Description

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

This class gather two known bugs:

1. GDCM 1.2.0 would rewrite [VR](#)=UN [Value](#) Length on 2 bytes instead of 4 bytes
2. GDCM 1.2.0 would also rewrite [DataElement](#) as Implicit when the [VR](#) would not be known this would only happen in some very rare cases. gdcm 2.X design could handle bug #1 or #2 exclusively, this class can now handle file which have both issues. See: [gdcmData/TheralysGDCM120Bug.dcm](#)

12.363.2 Member Function Documentation

12.363.2.1 GetLength()

[VL](#) [gdcm::UNExplicitImplicitDataElement::GetLength](#) () const

12.363.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::Read (
    std::istream & is)
```

12.363.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.363.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::ReadValue (
    std::istream & is)
```

The documentation for this class was generated from the following file:

- [gdcmUNExplicitImplicitDataElement.h](#)

12.364 gdcm::Unpacker12Bits Class Reference

Pack/Unpack 12 bits pixel into 16bits.

```
#include <gdcmUnpacker12Bits.h>
```

Static Public Member Functions

- static bool [Pack](#) (char *out, const char *in, size_t n)
- static bool [Unpack](#) (char *out, const char *in, size_t n)

12.364.1 Detailed Description

Pack/Unpack 12 bits pixel into 16bits.

- You can only pack an even number of 16bits, which means a multiple of 4 (expressed in bytes)
- You can only unpack a multiple of 3 bytes

This class has no purpose in general purpose DICOM implementation. However to be able to cope with some early ACR-NEMA file generated by a well-known private vendor, one would need to unpack 12bits Stored Pixel [Value](#) into a more standard 16bits Stored Pixel [Value](#).

See also

[Rescaler](#)

12.364.2 Member Function Documentation

12.364.2.1 Pack()

```
bool gdcm::Unpacker12Bits::Pack (  
    char * out,  
    const char * in,  
    size_t n) [static]
```

Pack an array of 16bits where all values are 12bits into a pack form. n is the length in bytes of array in, out will be a fake 8bits array of size $(n / 2) * 3$

12.364.2.2 Unpack()

```
bool gdcmm::Unpacker12Bits::Unpack (  
    char * out,  
    const char * in,  
    size_t n) [static]
```

Unpack an array of 'packed' 12bits data into a more conventional 16bits array. n is the length in bytes of array in, out will be a 16bits array of size $(n / 3) * 2$

The documentation for this class was generated from the following file:

- [gdcmmUnpacker12Bits.h](#)

12.365 gdcmm::Usage Class Reference

[Usage](#).

```
#include <gdcmmUsage.h>
```

Public Types

- enum [UsageType](#) {
 [Mandatory](#) ,
 [Conditional](#) ,
 [UserOption](#) ,
 [Invalid](#) }

Public Member Functions

- [Usage](#) ([UsageType](#) type=[Invalid](#))
- [operator UsageType](#) () const

Static Public Member Functions

- static const char * [GetUsageString](#) ([UsageType](#) type)
- static [UsageType](#) [GetUsageType](#) (const char *type)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Usage](#) &vr)

12.365.1 Detailed Description

Usage.

Note

A.1.3 **IOD Module Table** and Functional Group **Macro Table** This Section of each **IOD** defines in a tabular form the **Modules** comprising the **IOD**. The following information must be specified for each **Module** in the table:

- The name of the **Module** or Functional Group
- A reference to the Section in Annex C which defines the **Module** or Functional Group
- The usage of the **Module** or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The **Modules** referenced are defined in Annex C. A.↔1.3.1 MANDATORY MODULES For each **IOD**, Mandatory **Modules** shall be supported per the definitions, semantics and requirements defined in Annex C.

A.1.3.2 CONDITIONAL MODULES Conditional **Modules** are Mandatory **Modules** if specific conditions are met. If the specified conditions are not met, this **Module** shall not be supported; that is, no information defined in that **Module** shall be sent. A.1.3.3 USER OPTION MODULES User Option **Modules** may or may not be supported. If an optional **Module** is supported, the **Attribute** Types specified in the **Modules** in Annex C shall be supported.

12.365.2 Member Enumeration Documentation

12.365.2.1 UsageType

```
enum gdcm::Usage::UsageType
```

Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

12.365.3 Constructor & Destructor Documentation

12.365.3.1 Usage()

```
gdcm::Usage::Usage (
    UsageType type = Invalid) [inline]
```

References [Invalid](#).

Referenced by [operator<<](#).

12.365.4 Member Function Documentation

12.365.4.1 GetUsageString()

```
const char * gdcmm::Usage::GetUsageString (  
    UsageType type) [static]
```

Referenced by [operator<<](#).

12.365.4.2 GetUsageType()

```
UsageType gdcmm::Usage::GetUsageType (  
    const char * type) [static]
```

12.365.4.3 operator UsageType()

```
gdcmm::Usage::operator UsageType () const [inline]
```

12.365.5 Friends And Related Symbol Documentation

12.365.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Usage & vr) [friend]
```

References [Usage\(\)](#), and [GetUsageString\(\)](#).

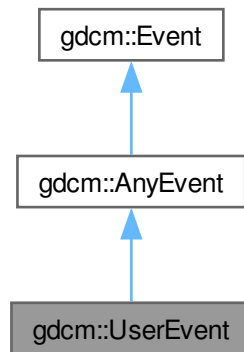
The documentation for this class was generated from the following file:

- [gdcmmUsage.h](#)

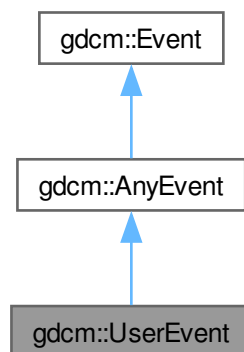
12.366 gdcm::UserEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



Additional Inherited Members

Public Member Functions inherited from [gdcm::Event](#)

- [Event](#) ()

- [Event](#) (const Event &)
- virtual [~Event](#) ()
- virtual bool [CheckEvent](#) (const [Event](#) *) const =0
- virtual const char * [GetEventName](#) () const =0
- virtual [Event](#) * [MakeObject](#) () const =0
- void [operator=](#) (const [Event](#) &)=delete
- virtual void [Print](#) (std::ostream &os) const

The documentation for this class was generated from the following file:

- [gdcmEvent.h](#)

12.367 gdcm::network::UserInformation Class Reference

[UserInformation.](#)

```
#include <gdcmUserInformation.h>
```

Public Member Functions

- [UserInformation](#) ()
- [UserInformation](#) (const UserInformation &)=delete
- [~UserInformation](#) ()
- void [AddRoleSelectionSub](#) ([RoleSelectionSub](#) const &r)
- void [AddSOPClassExtendedNegociationSub](#) ([SOPClassExtendedNegociationSub](#) const &s)
- [MaximumLengthSub](#) & [GetMaximumLengthSub](#) ()
- const [MaximumLengthSub](#) & [GetMaximumLengthSub](#) () const
- [UserInformation](#) & [operator=](#) (const [UserInformation](#) &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

12.367.1 Detailed Description

[UserInformation.](#)

[Table 9-16](#) USER INFORMATION ITEM FIELDS

TODO what is the goal of :

[Table 9-20](#) USER INFORMATION ITEM FIELDS

12.367.2 Constructor & Destructor Documentation

12.367.2.1 UserInformation() [1/2]

gdcm::network::UserInformation::UserInformation ()

Referenced by [UserInformation\(\)](#), and [operator=\(\)](#).

12.367.2.2 ~UserInformation()

gdcm::network::UserInformation::~~UserInformation ()

12.367.2.3 UserInformation() [2/2]

gdcm::network::UserInformation::UserInformation (
 const UserInformation &) [delete]

References [UserInformation\(\)](#).

12.367.3 Member Function Documentation

12.367.3.1 AddRoleSelectionSub()

void gdcm::network::UserInformation::AddRoleSelectionSub (
 [RoleSelectionSub](#) const & r)

12.367.3.2 AddSOPClassExtendedNegociationSub()

void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (
 [SOPClassExtendedNegociationSub](#) const & s)

12.367.3.3 GetMaximumLengthSub() [1/2]

[MaximumLengthSub](#) & gdcm::network::UserInformation::GetMaximumLengthSub () [inline]

12.367.3.4 GetMaximumLengthSub() [2/2]

const [MaximumLengthSub](#) & gdcm::network::UserInformation::GetMaximumLengthSub () const [inline]

12.367.3.5 operator=()

```
UserInformation & gdcmm::network::UserInformation::operator= (  
    const UserInformation & )
```

References [UserInformation\(\)](#).

12.367.3.6 Print()

```
void gdcmm::network::UserInformation::Print (  
    std::ostream & os) const
```

12.367.3.7 Read()

```
std::istream & gdcmm::network::UserInformation::Read (  
    std::istream & is)
```

12.367.3.8 Size()

```
size_t gdcmm::network::UserInformation::Size () const
```

12.367.3.9 Write()

```
const std::ostream & gdcmm::network::UserInformation::Write (  
    std::ostream & os) const
```

The documentation for this class was generated from the following file:

- [gdcmmUserInformation.h](#)

12.368 gdcmm::UUIDGenerator Class Reference

Class for generating unique UUID.

```
#include <gdcmmUUIDGenerator.h>
```

Public Member Functions

- const char * [Generate](#) ()

Static Public Member Functions

- static bool [IsValid](#) (const char *uid)
Find out if the string is a valid UUID or not.

12.368.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

12.368.2 Member Function Documentation

12.368.2.1 Generate()

```
const char * gdcm::UUIDGenerator::Generate ()
```

Return the generated uuid NOT THREAD SAFE

12.368.2.2 IsValid()

```
bool gdcm::UUIDGenerator::IsValid (  
    const char * uid) [static]
```

Find out if the string is a valid UUID or not.

The documentation for this class was generated from the following file:

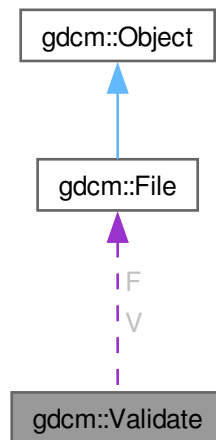
- [gdcmUUIDGenerator.h](#)

12.369 gdcm::Validate Class Reference

[Validate](#) class.

```
#include <gdcmValidate.h>
```

Collaboration diagram for `gdcM::Validate`:



Public Member Functions

- [Validate](#) ()
- [~Validate](#) ()
- const [File](#) & [GetValidatedFile](#) ()
- void [SetFile](#) ([File](#) const &f)
- void [Validation](#) ()

Protected Attributes

- const [File](#) * [F](#)
- [File](#) [V](#)

12.369.1 Detailed Description

[Validate](#) class.

12.369.2 Constructor & Destructor Documentation

12.369.2.1 Validate()

`gdcM::Validate::Validate ()`

12.369.2.2 ~Validate()

gdcm::Validate::~~Validate ()

12.369.3 Member Function Documentation

12.369.3.1 GetValidatedFile()

const [File](#) & gdcm::Validate::GetValidatedFile () [inline]

References [V](#).

12.369.3.2 SetFile()

void gdcm::Validate::SetFile (
 [File](#) const & f) [inline]

References [F](#).

12.369.3.3 Validation()

void gdcm::Validate::Validation ()

12.369.4 Member Data Documentation

12.369.4.1 F

const [File](#)* gdcm::Validate::F [protected]

Referenced by [SetFile\(\)](#).

12.369.4.2 V

[File](#) gdcm::Validate::V [protected]

Referenced by [GetValidatedFile\(\)](#).

The documentation for this class was generated from the following file:

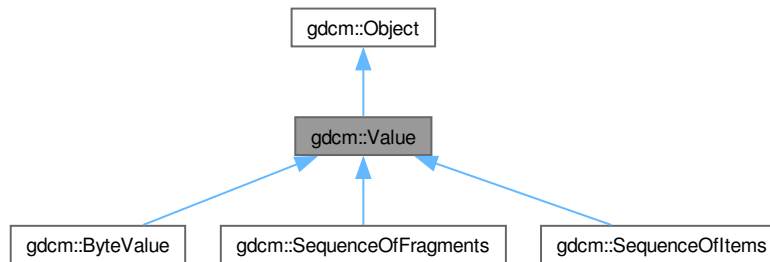
- [gdcmValidate.h](#)

12.370 gdcm::Value Class Reference

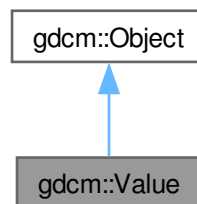
Class to represent the value of a Data [Element](#).

```
#include <gdcmValue.h>
```

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- `Value ()`=default
- `~Value ()` override=default
- virtual void `Clear ()`=0
- virtual `VL GetLength ()` const =0
- virtual bool `operator==` (const `Value` &val) const =0
- virtual void `SetLength (VL l)`=0

- `Object ()`
- `Object (const Object &)`
Special requirement for copy/cstor, assignment operator.
- `virtual ~Object ()`
- `void operator= (const Object &)`
- `virtual void Print (std::ostream &) const`

- virtual void `SetLengthOnly` (VL 1)

- void Register ()
- void UnRegister ()

- class **DataElement**

Class to represent the value of a Data [Element](#).

VALUE: A component of a Value Field. A Value Field may consist of one or more of these components.

12.370.2.1 Value()

Referenced by `gdcmm::ByteValue::operator==()`, `gdcmm::SequenceOfFragments::operator==()`, `gdcmm::SequenceOfItems::operator==()` and `operator==()`.

gdc::Value::~~Value () [override], [default]

12.370.3 Member Function Documentation

12.370.3.1 Clear()

virtual void gdcm::Value::Clear () [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

12.370.3.2 GetLength()

virtual [VL](#) gdcm::Value::GetLength () const [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), and [gdcm::DataElement::SetValue\(\)](#).

12.370.3.3 operator==()

virtual bool gdcm::Value::operator== (
const [Value](#) & val) const [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

References [Value\(\)](#).

12.370.3.4 SetLength()

virtual void gdcm::Value::SetLength (
[VL](#) l) [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfFragments](#), and [gdcm::SequenceOfItems](#).

12.370.3.5 SetLengthOnly()

virtual void gdcm::Value::SetLengthOnly (
[VL](#) l) [protected], [virtual]

Reimplemented in [gdcm::ByteValue](#).

12.370.4 Friends And Related Symbol Documentation

12.370.4.1 DataElement

friend class DataElement [friend]

References [DataElement](#).

Referenced by [DataElement](#).

The documentation for this class was generated from the following file:

- [gdcmValue.h](#)

12.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference

Class to dispatch template calls.

```
#include <gdcmValueIO.h>
```

Static Public Member Functions

- static std::istream & [Read](#) (std::istream &is, [Value](#) &v, bool readvalues)
- static const std::ostream & [Write](#) (std::ostream &os, const [Value](#) &v)

12.371.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>  
class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

12.371.2 Member Function Documentation

12.371.2.1 Read()

```
template<typename TDE, typename TSwap, typename TType = uint8_t>  
std::istream & gdcm::ValueIO< TDE, TSwap, TType >::Read (  
    std::istream & is,  
    Value & v,  
    bool readvalues) [static]
```

12.371.2.2 Write()

```
template<typename TDE, typename TSwap, typename TType = uint8_t>
const std::ostream & gdcmm::ValueIO< TDE, TSwap, TType >::Write (
    std::ostream & os,
    const Value & v) [static]
```

The documentation for this class was generated from the following file:

- [gdcmmValueIO.h](#)

12.372 gdcmm::MrProtocol::Vector3 Struct Reference

```
#include <gdcmmMrProtocol.h>
```

Public Attributes

- double [dCor](#)
- double [dSag](#)
- double [dTra](#)

12.372.1 Member Data Documentation

12.372.1.1 dCor

```
double gdcmm::MrProtocol::Vector3::dCor
```

12.372.1.2 dSag

```
double gdcmm::MrProtocol::Vector3::dSag
```

12.372.1.3 dTra

```
double gdcmm::MrProtocol::Vector3::dTra
```

The documentation for this struct was generated from the following file:

- [gdcmmMrProtocol.h](#)

12.373 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

Public Member Functions

- [Version](#) ()=default
- [~Version](#) ()=default
- void [Print](#) (std::ostream &os=std::cout) const

Static Public Member Functions

- static int [GetBuildVersion](#) ()
- static int [GetMajorVersion](#) ()
- static int [GetMinorVersion](#) ()
- static const char * [GetVersion](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Version](#) &v)

12.373.1 Detailed Description

major/minor and build version

12.373.2 Constructor & Destructor Documentation

12.373.2.1 Version()

gdcm::Version::Version () [default]

Referenced by [operator<<](#).

12.373.2.2 ~Version()

gdcm::Version::~~Version () [default]

12.373.3 Member Function Documentation

12.373.3.1 GetBuildVersion()

int gdcm::Version::GetBuildVersion () [static]

12.373.3.2 GetMajorVersion()

```
int gdcmm::Version::GetMajorVersion () [static]
```

12.373.3.3 GetMinorVersion()

```
int gdcmm::Version::GetMinorVersion () [static]
```

12.373.3.4 GetVersion()

```
const char * gdcmm::Version::GetVersion () [static]
```

12.373.3.5 Print()

```
void gdcmm::Version::Print (  
    std::ostream & os = std::cout) const
```

Referenced by [operator<<](#).

12.373.4 Friends And Related Symbol Documentation

12.373.4.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & __os,  
    const Version & v) [friend]
```

References [Version\(\)](#), and [Print\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmmVersion.h](#)

12.374 gdcmm::VL Class Reference

[Value](#) Length.

```
#include <gdcmmVL.h>
```

Public Types

- typedef uint32_t [Type](#)

Public Member Functions

- [VL](#) (uint32_t vl=0)
- [VL GetLength](#) () const
- bool [IsOdd](#) () const
Return whether or not the [VL](#) is odd or not.
- bool [IsUndefined](#) () const
- [operator uint32_t](#) () const
- [VL & operator++](#) ()
- [VL operator++](#) (int)
- [VL & operator+=](#) ([VL](#) const &vl)
+= operator
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [Read16](#) (std::istream &is)
- void [SetToUndefined](#) ()
- template<typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const
- template<typename TSwap>
const std::ostream & [Write16](#) (std::ostream &os) const

Static Public Member Functions

- static uint16_t [GetVL16Max](#) ()
- static uint32_t [GetVL32Max](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [VL](#) &vl)

12.374.1 Detailed Description

[Value](#) Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples

[BasicImageAnonymizer.cs](#), [DecompressImage.cs](#), [ReadAndDumpDICOMDIR2.cxx](#), and [rle2img.cxx](#).

12.374.2 Member Typedef Documentation

12.374.2.1 Type

typedef uint32_t [gdcm::VL::Type](#)

12.374.3 Constructor & Destructor Documentation

12.374.3.1 VL()

```
gdcm::VL::VL (  
    uint32_t vl = 0) [inline]
```

Referenced by [GetLength\(\)](#), [operator++\(\)](#), [operator++\(\)](#), [operator+=\(\)](#), and [operator<<](#).

12.374.4 Member Function Documentation

12.374.4.1 GetLength()

```
VL gdcm::VL::GetLength () const [inline]
```

Examples

[ReadAndDumpDICOMDIR2.cxx](#).

References [VL\(\)](#).

Referenced by [gdcm::FileMetaInformation::GetFullLength\(\)](#), and [gdcm::Item::Write\(\)](#).

12.374.4.2 GetVL16Max()

```
uint16_t gdcm::VL::GetVL16Max () [inline], [static]
```

12.374.4.3 GetVL32Max()

```
uint32_t gdcm::VL::GetVL32Max () [inline], [static]
```

12.374.4.4 IsOdd()

```
bool gdcm::VL::IsOdd () const [inline]
```

Return whether or not the [VL](#) is odd or not.

References [IsUndefined\(\)](#).

Referenced by [Write\(\)](#), and [Write16\(\)](#).

12.374.4.5 IsUndefined()

```
bool gdcv::VL::IsUndefined () const [inline]
```

Referenced by [IsOdd\(\)](#).

12.374.4.6 operator uint32_t()

```
gdcv::VL::operator uint32_t () const [inline]
```

12.374.4.7 operator++() [1/2]

```
VL & gdcv::VL::operator++ () [inline]
```

References [VL\(\)](#).

12.374.4.8 operator++() [2/2]

```
VL gdcv::VL::operator++ (  
    int ) [inline]
```

References [VL\(\)](#).

12.374.4.9 operator+=()

```
VL & gdcv::VL::operator+= (  
    VL const & vl) [inline]
```

+= operator

References [VL\(\)](#).

12.374.4.10 Read()

```
template<typename TSwap>  
std::istream & gdcv::VL::Read (  
    std::istream & is) [inline]
```

12.374.4.11 Read16()

```
template<typename TSwap>  
std::istream & gdcv::VL::Read16 (  
    std::istream & is) [inline]
```

References [gdcv_assert](#).

12.374.4.12 SetToUndefined()

```
void gdcml::VL::SetToUndefined () [inline]
```

12.374.4.13 Write()

```
template<typename TSwap>
const std::ostream & gdcml::VL::Write (
    std::ostream & os) const [inline]
```

References [IsOdd\(\)](#).

Referenced by [gdcml::Fragment::Write\(\)](#), [gdcml::Item::Write\(\)](#), [gdcml::SequenceOfFragments::Write\(\)](#), and [gdcml::SequenceOfItems::Write\(\)](#).

12.374.4.14 Write16()

```
template<typename TSwap>
const std::ostream & gdcml::VL::Write16 (
    std::ostream & os) const [inline]
```

References [gdcml_assert](#), and [IsOdd\(\)](#).

12.374.5 Friends And Related Symbol Documentation

12.374.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const VL & vl) [friend]
```

References [VL\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmlVL.h](#)

12.375 gdcml::VM Class Reference

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

```
#include <gdcmlVM.h>
```

Public Types

- enum [VMType](#) {
 - [VM0](#) = 0 ,
 - [VM1](#) = 1 ,
 - [VM2](#) = 2 ,
 - [VM3](#) = 4 ,
 - [VM4](#) = 8 ,
 - [VM5](#) = 16 ,
 - [VM6](#) = 32 ,
 - [VM8](#) = 64 ,
 - [VM9](#) = 128 ,
 - [VM10](#) = 256 ,
 - [VM12](#) = 512 ,
 - [VM16](#) = 1024 ,
 - [VM18](#) = 2048 ,
 - [VM24](#) = 4096 ,
 - [VM28](#) = 8192 ,
 - [VM32](#) = 16384 ,
 - [VM35](#) = 32768 ,
 - [VM99](#) = 65536 ,
 - [VM256](#) = 131072 ,
 - [VM1_2](#) = VM1 | VM2 ,
 - [VM1_3](#) = VM1 | VM2 | VM3 ,
 - [VM1_4](#) = VM1 | VM2 | VM3 | VM4 ,
 - [VM1_5](#) = VM1 | VM2 | VM3 | VM4 | VM5 ,
 - [VM1_8](#) = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 ,
 - [VM1_32](#) = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 ,
 - [VM1_99](#) = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 ,
 - [VM1_n](#) = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
 - [VM2_2n](#) = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256 ,
 - [VM2_n](#) = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
 - [VM3_4](#) = VM3 | VM4 ,
 - [VM3_3n](#) = VM3 | VM6 | VM9 | VM24 | VM99 | VM256 ,
 - [VM3_n](#) = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
 - [VM4_4n](#) = VM4 | VM16 | VM24 | VM32 | VM256 ,
 - [VM6_6n](#) = VM6 | VM12 | VM18 | VM24 ,
 - [VM6_n](#) = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
 - [VM7_7n](#) ,
 - [VM30_30n](#) ,
 - [VM47_47n](#) ,
 - [VM_END](#) = VM1_n + 1 }

Public Member Functions

- [VM](#) ([VMType](#) type=[VM0](#))
- bool [Compatible](#) ([VM](#) const &vm) const
- unsigned int [GetLength](#) () const
- operator [VMType](#) () const

Static Public Member Functions

- static size_t [GetNumberOfElementsFromArray](#) (const char *array, size_t length)
- static const char * [GetVMString](#) (VMType vm)
- static VMType [GetVMType](#) (const char *vm)
- static VMType [GetVMTypeFromLength](#) (size_t length, unsigned int size)
- static bool [IsValid](#) (int vm1, VMType vm2)

Static Protected Member Functions

- static unsigned int [GetIndex](#) (VMType vm)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const VM &vm)

12.375.1 Detailed Description

[Value](#) Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Some private dict define some more: 4-4n 1-4 1-5 256 9 3-4

even more:

7-7n 10 18 12 35 47_47n 30_30n 28

6-6n

12.375.2 Member Enumeration Documentation

12.375.2.1 VMType

enum [gdcm::VM::VMType](#)

Enumerator

VM0	
VM1	
VM2	
VM3	
VM4	
VM5	

VM6	
VM8	
VM9	
VM10	
VM12	
VM16	
VM18	
VM24	
VM28	
VM32	
VM35	
VM99	
VM256	
VM1_2	
VM1_3	
VM1_4	
VM1_5	
VM1_8	
VM1_32	
VM1_99	
VM1_n	
VM2_2n	
VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM6_n	
VM7_7n	
VM30_30n	
VM47_47n	

VM_END	
--------	--

12.375.3 Constructor & Destructor Documentation

12.375.3.1 VM()

```
gdcmm::VM::VM (
    VMType type = VM0) [inline]
```

References [VM0](#).

Referenced by [Compatible\(\)](#), [GetLength\(\)](#), and [operator<<](#).

12.375.4 Member Function Documentation

12.375.4.1 Compatible()

```
bool gdcmm::VM::Compatible (
    VM const & vm) const
```

WARNING: Implementation deficiency The Compatible function is poorly implemented, the reference vm should be coming from the dictionary, while the passed in value is the value guess from the file.

References [VM\(\)](#).

12.375.4.2 GetIndex()

```
unsigned int gdcmm::VM::GetIndex (
    VMType vm) [static], [protected]
```

12.375.4.3 GetLength()

```
unsigned int gdcmm::VM::GetLength () const
```

References [VM\(\)](#), and [operator<<](#).

12.375.4.4 GetNumberOfElementsFromArray()

```
size_t gdcmm::VM::GetNumberOfElementsFromArray (
    const char * array,
    size_t length) [static]
```

12.375.4.5 GetVMString()

```
const char * gdcm::VM::GetVMString (
    VMType vm) [static]
```

Return the string as written in the official DICOM dict from a custom enum type

Referenced by [operator<<](#).

12.375.4.6 GetVMType()

```
VMType gdcm::VM::GetVMType (
    const char * vm) [static]
```

12.375.4.7 GetVMTypeFromLength()

```
VMType gdcm::VM::GetVMTypeFromLength (
    size_t length,
    unsigned int size) [static]
```

12.375.4.8 IsValid()

```
bool gdcm::VM::IsValid (
    int vm1,
    VMType vm2) [static]
```

Check if vm1 is valid compare to vm2, i.e vm1 is element of vm2 vm1 is typically deduce from counting in a ValueField

12.375.4.9 operator VMType()

```
gdcm::VM::operator VMType () const [inline]
```

12.375.5 Friends And Related Symbol Documentation

12.375.5.1 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const VM & vm) [friend]
```

References [VM\(\)](#), [gdcm_assert](#), and [GetVMString\(\)](#).

Referenced by [GetLength\(\)](#).

The documentation for this class was generated from the following file:

- [gdcmVM.h](#)

12.376 gdcm::VMToLength< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- [gdcmVM.h](#)

12.377 gdcm::VR Class Reference

[VR](#) class.

```
#include <gdcmVR.h>
```

Public Types

- enum [VRType](#) : long long {
 [INVALID](#) = 0 ,
 [AE](#) = 1 ,
 [AS](#) = 2 ,
 [AT](#) = 4 ,
 [CS](#) = 8 ,
 [DA](#) = 16 ,
 [DS](#) = 32 ,
 [DT](#) = 64 ,
 [FD](#) = 128 ,
 [FL](#) = 256 ,
 [IS](#) = 512 ,
 [LO](#) = 1024 ,
 [LT](#) = 2048 ,
 [OB](#) = 4096 ,
 [OD](#) = 134217728 ,
 [OF](#) = 8192 ,
 [OL](#) = 268435456 ,
 [OV](#) = 2147483648 ,
 [OW](#) = 16384 ,
 [PN](#) = 32768 ,
 [SH](#) = 65536 ,
 [SL](#) = 131072 ,
 [SQ](#) = 262144 ,
 [SS](#) = 524288 ,
 [ST](#) = 1048576 ,
 [SV](#) = 4294967296 ,
 [TM](#) = 2097152 ,
 [UC](#) = 536870912 ,
 [UI](#) = 4194304 ,
 [UL](#) = 8388608 ,
 [UN](#) = 16777216 ,
 [UR](#) = 1073741824 ,
 [US](#) = 33554432 ,
 [UT](#) = 67108864 ,

```

UV = 8589934592 ,
OB_OW = OB | OW ,
US_SS = US | SS ,
US_SS_OW = US | SS | OW ,
US_OW = US | OW ,
VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM |
UI | UL | US ,
VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV ,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT ,
VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US |
UV ,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN ,
VRALL = VRASCII | VRBINARY ,
VR_END = UV+1 }

```

Public Member Functions

- [VR](#) ([VRType](#) vr=[INVALID](#))
- bool [Compatible](#) ([VR](#) const &vr) const
- int [GetLength](#) () const
- unsigned int [GetSize](#) () const
- unsigned int [GetSizeof](#) () const
- bool [IsDual](#) () const
- bool [IsVRFile](#) () const
- operator [VRType](#) () const
- std::istream & [Read](#) (std::istream &is)
- const std::ostream & [Write](#) (std::ostream &os) const

Static Public Member Functions

- static bool [CanDisplay](#) ([VRType](#) vr)
- static uint32_t [GetLength](#) ([VRType](#) vr)
- static const char * [GetVRString](#) ([VRType](#) vr)
- static const char * [GetVRStringFromFile](#) ([VRType](#) vr)
- static [VRType](#) [GetVRType](#) (const char *vr)
- static [VRType](#) [GetVRTypeFromFile](#) (const char *vr)
- static bool [IsASCII](#) ([VRType](#) vr)
- static bool [IsASCII2](#) ([VRType](#) vr)
- static bool [IsBinary](#) ([VRType](#) vr)
- static bool [IsBinary2](#) ([VRType](#) vr)
- static bool [IsSwap](#) (const char *vr)
- static bool [IsValid](#) (const char *vr)
- static bool [IsValid](#) (const char *vr1, [VRType](#) vr2)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [VR](#) &vr)

12.377.1 Detailed Description

[VR](#) class.

This is adapted from DICOM standard The biggest difference is the INVALID [VR](#) and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict

Note

VALUE REPRESENTATION ([VR](#)) Specifies the data type and format of the Value(s) contained in the [Value](#) Field of a Data [Element](#). VALUE REPRESENTATION FIELD: The field where the [Value](#) Representation of a Data [Element](#) is stored in the encoding of a Data [Element](#) structure with explicit [VR](#).

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [SimplePrint.cs](#).

12.377.2 Member Enumeration Documentation

12.377.2.1 VRType

enum [gdcm::VR::VRType](#) : long long

Enumerator

INVALID	
AE	
AS	
AT	
CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	

OL	
OV	
OW	
PN	
SH	
SL	
SQ	
SS	
ST	
SV	
TM	
UC	
UI	
UL	
UN	
UR	
US	
UT	
UV	
OB_OW	
US_SS	
US_SS_OW	
US_OW	
VL16	
VL32	
VRASCII	
VRBINARY	
VR_VM1	
VRALL	
VR_END	

Examples

[NewSequence.cs](#), and [SimplePrint.cs](#).

12.377.3 Constructor & Destructor Documentation

12.377.3.1 VR()

```
gdcmm::VR::VR (
    VRType vr = INVALID) [inline]
```

References [INVALID](#).

Referenced by [Compatible\(\)](#), and [operator<<](#).

12.377.4 Member Function Documentation

12.377.4.1 CanDisplay()

```
bool gdcmm::VR::CanDisplay (  
    VRType vr) [static]
```

12.377.4.2 Compatible()

```
bool gdcmm::VR::Compatible (  
    VR const & vr) const
```

Examples

[SimplePrint.cs](#).

References [VR\(\)](#).

12.377.4.3 GetLength() [1/2]

```
int gdcmm::VR::GetLength () const [inline]
```

References [GetLength\(\)](#).

Referenced by [GetLength\(\)](#).

12.377.4.4 GetLength() [2/2]

```
uint32_t gdcmm::VR::GetLength (  
    VRType vr) [inline], [static]
```

References [VL32](#).

12.377.4.5 GetSize()

```
unsigned int gdcmm::VR::GetSize () const [inline]
```

References [AE](#), [AS](#), [AT](#), [CS](#), [DA](#), [DS](#), [DT](#), [FD](#), [FL](#), [gdcmm_assert](#), [INVALID](#), [IS](#), [LO](#), [LT](#), [OB](#), [OB_OW](#), [OD](#), [OF](#), [OL](#), [OV](#), [OW](#), [PN](#), [SH](#), [SL](#), [SQ](#), [SS](#), [ST](#), [SV](#), [TM](#), [UC](#), [UI](#), [UL](#), [UN](#), [UR](#), [US](#), [US_OW](#), [US_SS](#), [US_SS_OW](#), [UT](#), [UV](#), [VL16](#), [VL32](#), [VR_END](#), [VR_VM1](#), [VRALL](#), [VRASCII](#), [VRBINARY](#), and [VRTypeTemplateCase](#).

12.377.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof () const
```

12.377.4.7 GetVRString()

```
const char * gdcm::VR::GetVRString (  
    VRType vr) [static]
```

Referenced by [operator<<](#), and [Write\(\)](#).

12.377.4.8 GetVRStringFromFile()

```
const char * gdcm::VR::GetVRStringFromFile (  
    VRType vr) [static]
```

12.377.4.9 GetVRType()

```
VRType gdcm::VR::GetVRType (  
    const char * vr) [static]
```

12.377.4.10 GetVRTypeFromFile()

```
VRType gdcm::VR::GetVRTypeFromFile (  
    const char * vr) [static]
```

Referenced by [Read\(\)](#).

12.377.4.11 IsASCII()

```
bool gdcm::VR::IsASCII (  
    VRType vr) [static]
```

12.377.4.12 IsASCII2()

```
bool gdcm::VR::IsASCII2 (  
    VRType vr) [static]
```

12.377.4.13 IsBinary()

```
bool gdcm::VR::IsBinary (  
    VRType vr) [static]
```

12.377.4.14 IsBinary2()

```
bool gdcm::VR::IsBinary2 (  
    VRType vr) [static]
```

12.377.4.15 IsDual()

```
bool gdcm::VR::IsDual () const
```

Referenced by [Write\(\)](#).

12.377.4.16 IsSwap()

```
bool gdcm::VR::IsSwap (  
    const char * vr) [static]
```

12.377.4.17 IsValid() [1/2]

```
bool gdcm::VR::IsValid (  
    const char * vr) [static]
```

12.377.4.18 IsValid() [2/2]

```
bool gdcm::VR::IsValid (  
    const char * vr1,  
    VRType vr2) [static]
```

12.377.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile () const
```

Referenced by [gdcm::DataElement::SetVR\(\)](#).

12.377.4.20 operator VRType()

```
gdcm::VR::operator VRType () const [inline]
```

12.377.4.21 Read()

```
std::istream & gdcm::VR::Read (  
    std::istream & is) [inline]
```

References [gdcm_assert](#), [gdcmDebugMacro](#), [GetVRTypeFromFile\(\)](#), [INVALID](#), [VL32](#), and [VR_END](#).

12.377.4.22 Write()

```
const std::ostream & gdcm::VR::Write (  
    std::ostream & os) const    [inline]
```

References [gdcm_assert](#), [gdcmAssertAlwaysMacro](#), [GetVRString\(\)](#), [INVALID](#), [IsDual\(\)](#), and [VL32](#).

12.377.5 Friends And Related Symbol Documentation

12.377.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const VR & vr)    [friend]
```

References [VR\(\)](#), and [GetVRString\(\)](#).

The documentation for this class was generated from the following file:

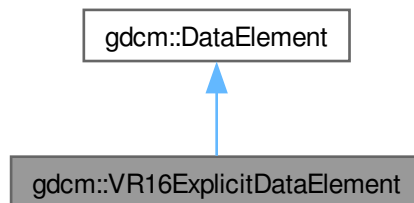
- [gdcmVR.h](#)

12.378 gdcm::VR16ExplicitDataElement Class Reference

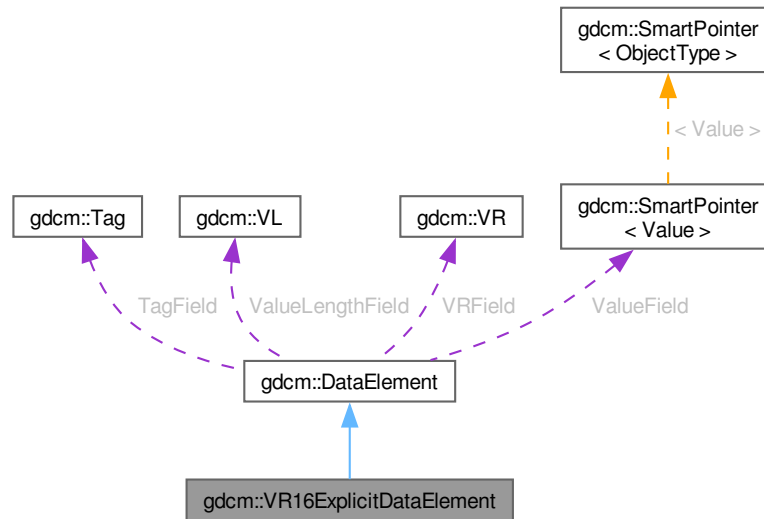
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#include <gdcmVR16ExplicitDataElement.h>
```

Inheritance diagram for gdcm::VR16ExplicitDataElement:



Collaboration diagram for `gdcm::VR16ExplicitDataElement`:



Public Member Functions

- [VL GetLength](#) () const
- template<typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Public Member Functions inherited from [gdcm::DataElement](#)

- [DataElement](#) (const [DataElement](#) &_val)
- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- void [Clear](#) ()
Clear [Data Element](#) (make [Value](#) empty and invalidate [Tag](#) + [VR](#)).
- void [Empty](#) ()
Make [Data Element](#) empty (no [Value](#)).
- const [ByteValue](#) * [GetByteValue](#) () const
- template<typename TDE>
[VL GetLength](#) () const
- [SequenceOfFragments](#) * [GetSequenceOfFragments](#) ()
- const [SequenceOfFragments](#) * [GetSequenceOfFragments](#) () const

- [Tag](#) & [GetTag](#) ()
- const [Tag](#) & [GetTag](#) () const
Get [Tag](#).
- [Value](#) & [GetValue](#) ()
- [Value](#) const & [GetValue](#) () const
Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- [VL](#) & [GetVL](#) ()
- const [VL](#) & [GetVL](#) () const
Get [VL](#).
- [VR](#) const & [GetVR](#) () const
- bool [IsEmpty](#) () const
Check if Data [Element](#) is empty.
- bool [IsUndefinedLength](#) () const
return if [Value](#) Length if of undefined length
- bool [operator](#)< (const [DataElement](#) &de) const
- [DataElement](#) & [operator](#)= (const [DataElement](#) &)=default
- bool [operator](#)== (const [DataElement](#) &de) const
- template<typename TDE, typename TSwap>
std::istream & [Read](#) (std::istream &is)
- template<typename TDE, typename TSwap>
std::istream & [ReadOrSkip](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadPreValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValue](#) (std::istream &is, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadValueWithLength](#) (std::istream &is, [VL](#) &length, std::set< [Tag](#) > const &skiptags)
- template<typename TDE, typename TSwap>
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)
- void [SetByteValue](#) (const char *array, [VL](#) length)
- void [SetTag](#) (const [Tag](#) &t)
- void [SetValue](#) ([Value](#) const &vl)
- void [SetVL](#) (const [VL](#) &vl)
- void [SetVLToUndefined](#) ()
- void [SetVR](#) ([VR](#) const &vr)
- template<typename TDE, typename TSwap>
const std::ostream & [Write](#) (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from [gdcm::DataElement](#)

- typedef [SmartPointer](#)< [Value](#) > [ValuePtr](#)

Protected Member Functions inherited from [gdcm::DataElement](#)

- void [SetValueFieldLength](#) ([VL](#) vl, bool readvalues)

Protected Attributes inherited from [gdcm::DataElement](#)

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

12.378.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Note

This class support 16 bits when finding an unknown [VR](#): For instance: Siemens_CT_Sensation64_↔
has_VR_RT.dcm

12.378.2 Member Function Documentation

12.378.2.1 GetLength()

[VL](#) [gdcm::VR16ExplicitDataElement::GetLength](#) () const

12.378.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::Read (
    std::istream & is)
```

12.378.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadPreValue (
    std::istream & is)
```

12.378.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true)
```

12.378.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length)
```

The documentation for this class was generated from the following file:

- [gdcmVR16ExplicitDataElement.h](#)

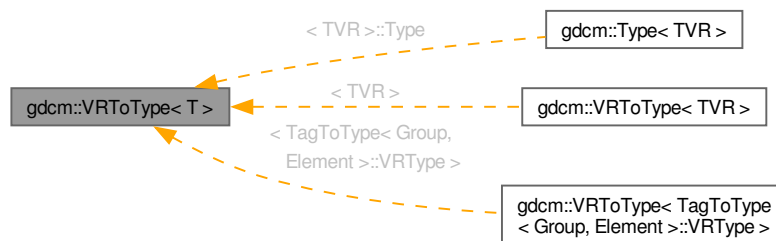
12.379 gdcm::VRToEncoding< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

12.380 gdcm::VRToType< T > Struct Template Reference

Inheritance diagram for gdcm::VRToType< T >:



12.380.1 Detailed Description

```
template<long long T>
struct gdcm::VRToType< T >
```

Examples

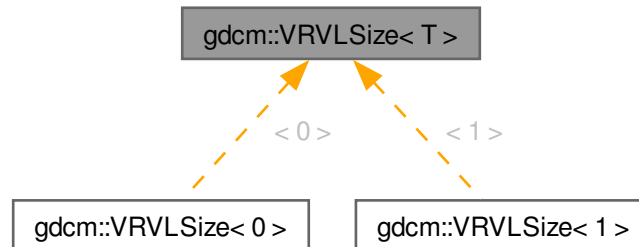
[DumpGEMSMovieGroup.cxx](#).

The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

12.381 gdc::VRVLSize< T > Class Template Reference

Inheritance diagram for gdc::VRVLSize< T >:



The documentation for this class was generated from the following file:

- [gdcAttribute.h](#)

12.382 gdc::VRVLSize< 0 > Class Reference

```
#include <gdcAttribute.h>
```

Inheritance diagram for gdc::VRVLSize< 0 >:



Collaboration diagram for gdcm::VRVLSIZE< 0 >:



Static Public Member Functions

- static uint16_t [Read](#) (std::istream &_is)
- static void [Write](#) (std::ostream &os)

12.382.1 Member Function Documentation

12.382.1.1 Read()

```
uint16_t gdcm::VRVLSIZE< 0 >::Read (  
    std::istream & _is) [inline], [static]
```

12.382.1.2 Write()

```
void gdcm::VRVLSIZE< 0 >::Write (  
    std::ostream & os) [inline], [static]
```

The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

12.383 gdcM::VRVLSIZE< 1 > Class Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::VRVLSIZE< 1 >:



Collaboration diagram for gdcM::VRVLSIZE< 1 >:



Static Public Member Functions

- static uint32_t [Read](#) (std::istream &__is)
- static void [Write](#) (std::ostream &os)

12.383.1 Member Function Documentation

12.383.1.1 Read()

```
uint32_t gdcM::VRVLSIZE< 1 >::Read (
    std::istream & __is) [inline], [static]
```

12.383.1.2 Write()

```
void gdcm::VRVLSize< 1 >::Write (  
    std::ostream & os)    [inline], [static]
```

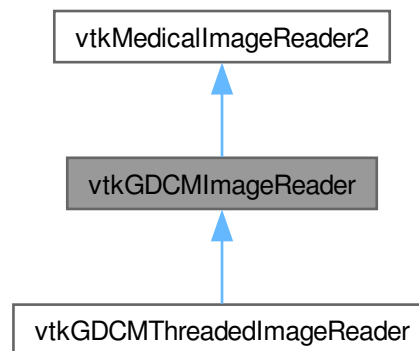
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

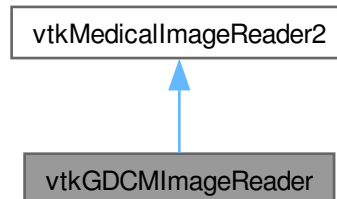
12.384 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Collaboration diagram for vtkGDCMImageReader:



Public Member Functions

- virtual int [CanReadFile](#) (const char *fname)
- virtual const char * [GetDescriptiveName](#) ()
- virtual const char * [GetFileExtensions](#) ()
- vtkImageData * [GetIconImage](#) ()
- vtkImageData * [GetOverlay](#) (int i)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetCurve](#) (vtkPolyData *pd)
- virtual void [SetFileNames](#) (vtkStringArray *)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties *pd)
- [vtkBooleanMacro](#) ([ApplyLookupTable](#), int)
- int [vtkBooleanMacro](#) ([ApplyYBRToRGB](#), int)
- [vtkBooleanMacro](#) ([LoadIconImage](#), int)
- [vtkBooleanMacro](#) ([LoadOverlays](#), int)
- [vtkBooleanMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([ApplyLookupTable](#), int)
- [vtkGetMacro](#) ([ApplyYBRToRGB](#), int) [vtkSetMacro](#)([ApplyYBRToRGB](#)
- [vtkGetMacro](#) ([ImageFormat](#), int)
- [vtkGetMacro](#) ([LoadIconImage](#), int)
- [vtkGetMacro](#) ([LoadOverlays](#), int)
- [vtkGetMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([NumberOfIconImages](#), int)
- [vtkGetMacro](#) ([NumberOfOverlays](#), int)
- [vtkGetMacro](#) ([PlanarConfiguration](#), int)
- [vtkGetMacro](#) ([Scale](#), double)
- [vtkGetMacro](#) ([Shift](#), double)
- [vtkGetObjectMacro](#) ([Curve](#), vtkPolyData)
- [vtkGetObjectMacro](#) ([DirectionCosines](#), vtkMatrix4x4)
- [vtkGetObjectMacro](#) ([FileNames](#), vtkStringArray)
- [vtkGetObjectMacro](#) ([MedicalImageProperties](#), vtkMedicalImageProperties)
- [vtkGetVector3Macro](#) ([ImagePositionPatient](#), double)
- [vtkGetVector6Macro](#) ([ImageOrientationPatient](#), double)
- [vtkSetMacro](#) ([ApplyLookupTable](#), int)
- [vtkSetMacro](#) ([LoadIconImage](#), int)
- [vtkSetMacro](#) ([LoadOverlays](#), int)
- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkTypeMacro](#) ([vtkGDCMImageReader](#), vtkMedicalImageReader2)

Static Public Member Functions

- static [vtkGDCMImageReader](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject *out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcmm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char *filename, char *pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char *)
- void [SetFilePrefix](#) (const char *)
- [vtkGetStringMacro](#) (FilePattern)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkSetVector6Macro](#) (ImageOrientationPatient, double)

Protected Attributes

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- vtkPolyData * [Curve](#)
- vtkMatrix4x4 * [DirectionCosines](#)
- vtkStringArray * [FileNames](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- vtkMedicalImageProperties * [MedicalImageProperties](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

12.384.1 Detailed Description

Examples

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [MagnifyFile.cxx](#), [MetaImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), [RefCounting.cs](#), [gdcmmorthoplanes.cxx](#), [gdcmmreslice.cxx](#), [gdcmmtexture.cxx](#), [gdcmmvolume.cxx](#), and [offscreenimage.cxx](#).

12.384.2 Constructor & Destructor Documentation

12.384.2.1 vtkGDCMImageReader()

vtkGDCMImageReader::vtkGDCMImageReader () [protected]

Examples

[HelloActiviz2.cs](#).

References [vtkGDCMImageReader\(\)](#).

Referenced by [vtkGDCMImageReader\(\)](#), [~vtkGDCMImageReader\(\)](#), [New\(\)](#), [vtkGetStringMacro\(\)](#), [vtkTypeMacro\(\)](#), and [vtkGDCMThreadedImageReader::vtkTypeMacro\(\)](#).

12.384.2.2 ~vtkGDCMImageReader()

vtkGDCMImageReader::~~vtkGDCMImageReader () [protected]

References [vtkGDCMImageReader\(\)](#).

12.384.3 Member Function Documentation

12.384.3.1 CanReadFile()

virtual int vtkGDCMImageReader::CanReadFile (
 const char * fname) [virtual]

Examples

[AWTMedical3.java](#), and [MetaImageMD5Activiz.cs](#).

12.384.3.2 ExecuteData()

void vtkGDCMImageReader::ExecuteData (
 vtkDataObject * out) [protected]

12.384.3.3 ExecuteInformation()

void vtkGDCMImageReader::ExecuteInformation () [protected]

12.384.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
    const gdcm::ImageReader & reader) [protected]
```

References [FillMedicalImageInformation\(\)](#).

Referenced by [FillMedicalImageInformation\(\)](#).

12.384.3.5 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader::GetDescriptiveName () [inline], [virtual]
```

12.384.3.6 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader::GetFileExtensions () [inline], [virtual]
```

12.384.3.7 GetIconImage()

```
vtkImageData * vtkGDCMImageReader::GetIconImage ()
```

12.384.3.8 GetOverlay()

```
vtkImageData * vtkGDCMImageReader::GetOverlay (
    int i)
```

12.384.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen) [protected]
```

12.384.3.10 New()

```
vtkGDCMImageReader * vtkGDCMImageReader::New () [static]
```

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [MagnifyFile.cxx](#), [MetaImageMD5Activiz.cs](#), [RefCounting.cs](#), [gdcmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmtexture.cxx](#), [gdcmvolume.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

References [vtkGDCMImageReader\(\)](#).

12.384.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (  
    ostream & os,  
    vtkIndent indent) [virtual]
```

Reimplemented in [vtkGDCMThreadedImageReader](#).

12.384.3.12 RequestDataCompat()

```
int vtkGDCMImageReader::RequestDataCompat () [protected]
```

References [RequestDataCompat\(\)](#).

Referenced by [RequestDataCompat\(\)](#).

12.384.3.13 RequestInformationCompat()

```
int vtkGDCMImageReader::RequestInformationCompat () [protected]
```

References [RequestInformationCompat\(\)](#).

Referenced by [RequestInformationCompat\(\)](#).

12.384.3.14 SetCurve()

```
virtual void vtkGDCMImageReader::SetCurve (  
    vtkPolyData * pd) [virtual]
```

References [SetCurve\(\)](#).

Referenced by [SetCurve\(\)](#).

12.384.3.15 SetFileNames()

```
virtual void vtkGDCMImageReader::SetFileNames (  
    vtkStringArray * ) [virtual]
```

Examples

[AWTMedical3.java](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [ReadSeriesIntoVTK.java](#), and [gdcmothoplanes.cxx](#).

12.384.3.16 SetFilePattern()

```
void vtkGDCMImageReader::SetFilePattern (
    const char * )    [inline], [protected]
```

12.384.3.17 SetFilePrefix()

```
void vtkGDCMImageReader::SetFilePrefix (
    const char * )    [inline], [protected]
```

12.384.3.18 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd)    [virtual]
```

12.384.3.19 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

References [ApplyLookupTable](#).

12.384.3.20 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

References [ApplyYBRToRGB](#), and [vtkBooleanMacro\(\)](#).

12.384.3.21 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

References [LoadIconImage](#).

12.384.3.22 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

References [LoadOverlays](#).

Referenced by [vtkBooleanMacro\(\)](#).

12.384.3.23 vtkBooleanMacro() [5/5]

vtkGDCMImageReader::vtkBooleanMacro (
 [LossyFlag](#) ,
 int)

References [LossyFlag](#).

12.384.3.24 vtkGetMacro() [1/11]

vtkGDCMImageReader::vtkGetMacro (
 [ApplyLookupTable](#) ,
 int)

References [ApplyLookupTable](#).

12.384.3.25 vtkGetMacro() [2/11]

vtkGDCMImageReader::vtkGetMacro (
 [ApplyYBRToRGB](#) ,
 int)

References [ApplyYBRToRGB](#), and [vtkSetMacro\(\)](#).

12.384.3.26 vtkGetMacro() [3/11]

vtkGDCMImageReader::vtkGetMacro (
 [ImageFormat](#) ,
 int)

References [ImageFormat](#), and [vtkGetMacro\(\)](#).

12.384.3.27 vtkGetMacro() [4/11]

vtkGDCMImageReader::vtkGetMacro (
 [LoadIconImage](#) ,
 int)

References [LoadIconImage](#).

12.384.3.28 vtkGetMacro() [5/11]

vtkGDCMImageReader::vtkGetMacro (
 [LoadOverlays](#) ,
 int)

References [LoadOverlays](#).

Referenced by [vtkGetMacro\(\)](#), [vtkGetMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkGetMacro\(\)](#).

12.384.3.29 vtkGetMacro() [6/11]

vtkGDCMImageReader::vtkGetMacro (
 [LossyFlag](#) ,
 int)

References [LossyFlag](#).

12.384.3.30 vtkGetMacro() [7/11]

vtkGDCMImageReader::vtkGetMacro (
 [NumberOfIconImages](#) ,
 int)

References [NumberOfIconImages](#).

12.384.3.31 vtkGetMacro() [8/11]

vtkGDCMImageReader::vtkGetMacro (
 [NumberOfOverlays](#) ,
 int)

References [NumberOfOverlays](#).

12.384.3.32 vtkGetMacro() [9/11]

vtkGDCMImageReader::vtkGetMacro (
 [PlanarConfiguration](#) ,
 int)

References [PlanarConfiguration](#), and [vtkGetMacro\(\)](#).

12.384.3.33 vtkGetMacro() [10/11]

vtkGDCMImageReader::vtkGetMacro (
 [Scale](#) ,
 double)

References [Scale](#), and [vtkGetMacro\(\)](#).

12.384.3.34 vtkGetMacro() [11/11]

vtkGDCMImageReader::vtkGetMacro (
 [Shift](#) ,
 double)

References [Shift](#), and [vtkGetMacro\(\)](#).

12.384.3.35 vtkGetObjectMacro() [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

References [Curve](#), and [vtkGetObjectMacro\(\)](#).

12.384.3.36 vtkGetObjectMacro() [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

References [DirectionCosines](#).

Referenced by [vtkGetObjectMacro\(\)](#).

12.384.3.37 vtkGetObjectMacro() [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

References [FileNames](#).

12.384.3.38 vtkGetObjectMacro() [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

References [MedicalImageProperties](#).

12.384.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePattern ) [protected]
```

References [vtkGDCMImageReader\(\)](#).

12.384.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePrefix ) [protected]
```

12.384.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

References [ImagePositionPatient](#), and [vtkGetVector3Macro\(\)](#).

Referenced by [vtkGetVector3Macro\(\)](#).

12.384.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

References [ImageOrientationPatient](#), and [vtkGetVector6Macro\(\)](#).

Referenced by [vtkGetVector6Macro\(\)](#).

12.384.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

References [ApplyLookupTable](#).

12.384.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LoadIconImage ,
    int )
```

References [LoadIconImage](#).

12.384.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LoadOverlays ,
    int )
```

References [LoadOverlays](#).

Referenced by [vtkGetMacro\(\)](#).

12.384.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LossyFlag ,
    int )
```

References [LossyFlag](#).

12.384.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

References [ImageOrientationPatient](#), and [vtkSetVector6Macro\(\)](#).

Referenced by [vtkSetVector6Macro\(\)](#).

12.384.3.48 vtkTypeMacro()

```
vtkGDCMImageReader::vtkTypeMacro (
    vtkGDCMImageReader ,
    vtkMedicalImageReader2 )
```

References [vtkGDCMImageReader\(\)](#).

12.384.4 Member Data Documentation

12.384.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo [protected]
```

12.384.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable [protected]
```

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.384.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]
```

12.384.4.4 ApplyShiftScale

int vtkGDCMImageReader::ApplyShiftScale [protected]

12.384.4.5 ApplyYBRToRGB

int vtkGDCMImageReader::ApplyYBRToRGB [protected]

Referenced by [vtkBooleanMacro\(\)](#), and [vtkGetMacro\(\)](#).

12.384.4.6 Curve

vtkPolyData* vtkGDCMImageReader::Curve [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.384.4.7 DirectionCosines

vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.384.4.8 FileNames

vtkStringArray* vtkGDCMImageReader::FileNames [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.384.4.9 ForceRescale

int vtkGDCMImageReader::ForceRescale [protected]

12.384.4.10 IconDataScalarType

int vtkGDCMImageReader::IconDataScalarType [protected]

12.384.4.11 IconImageDataExtent

int vtkGDCMImageReader::IconImageDataExtent[6] [protected]

12.384.4.12 IconNumberOfScalarComponents

int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]

12.384.4.13 ImageFormat

int vtkGDCMImageReader::ImageFormat [protected]

Referenced by [vtkGetMacro\(\)](#).

12.384.4.14 ImageOrientationPatient

double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]

Referenced by [vtkGetVector6Macro\(\)](#), and [vtkSetVector6Macro\(\)](#).

12.384.4.15 ImagePositionPatient

double vtkGDCMImageReader::ImagePositionPatient[3] [protected]

Referenced by [vtkGetVector3Macro\(\)](#).

12.384.4.16 LoadIconImage

int vtkGDCMImageReader::LoadIconImage [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.384.4.17 LoadOverlays

int vtkGDCMImageReader::LoadOverlays [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.384.4.18 LossyFlag

int vtkGDCMImageReader::LossyFlag [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.384.4.19 MedicalImageProperties

vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.384.4.20 NumberOfIconImages

int vtkGDCMImageReader::NumberOfIconImages [protected]

Referenced by [vtkGetMacro\(\)](#).

12.384.4.21 NumberOfOverlays

int vtkGDCMImageReader::NumberOfOverlays [protected]

Referenced by [vtkGetMacro\(\)](#).

12.384.4.22 PlanarConfiguration

int vtkGDCMImageReader::PlanarConfiguration [protected]

Referenced by [vtkGetMacro\(\)](#).

12.384.4.23 Scale

double vtkGDCMImageReader::Scale [protected]

Referenced by [vtkGetMacro\(\)](#), and [vtkGDCMThreadedImageReader::vtkSetMacro\(\)](#).

12.384.4.24 Shift

double vtkGDCMImageReader::Shift [protected]

Referenced by [vtkGetMacro\(\)](#), and [vtkGDCMThreadedImageReader::vtkSetMacro\(\)](#).

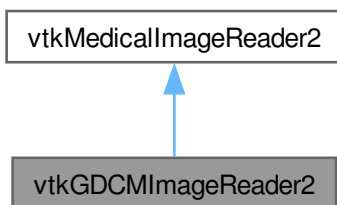
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

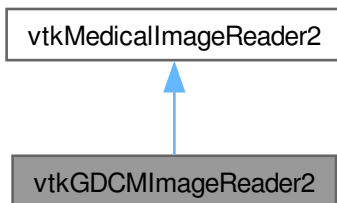
12.385 vtkGDCMImageReader2 Class Reference

```
#include <vtkGDCMImageReader2.h>
```

Inheritance diagram for vtkGDCMImageReader2:



Collaboration diagram for vtkGDCMImageReader2:



Public Member Functions

- virtual int [CanReadFile](#) (const char *fname)
- virtual const char * [GetDescriptiveName](#) ()
- virtual const char * [GetFileExtensions](#) ()
- vtkImageData * [GetIconImage](#) ()
- vtkAlgorithmOutput * [GetIconImagePort](#) ()
- vtkImageData * [GetOverlay](#) (int i)
- vtkAlgorithmOutput * [GetOverlayPort](#) (int index)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetCurve](#) (vtkPolyData *pd)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties *pd)
- [vtkBooleanMacro](#) ([ApplyLookupTable](#), int)

- int [vtkBooleanMacro](#) ([ApplyYBRToRGB](#), int)
- [vtkBooleanMacro](#) ([LoadIconImage](#), int)
- [vtkBooleanMacro](#) ([LoadOverlays](#), int)
- [vtkBooleanMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([ApplyLookupTable](#), int)
- [vtkGetMacro](#) ([ApplyYBRToRGB](#), int) [vtkSetMacro](#)([ApplyYBRToRGB](#)
- [vtkGetMacro](#) ([ImageFormat](#), int)
- [vtkGetMacro](#) ([LoadIconImage](#), int)
- [vtkGetMacro](#) ([LoadOverlays](#), int)
- [vtkGetMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([NumberOfIconImages](#), int)
- [vtkGetMacro](#) ([NumberOfOverlays](#), int)
- [vtkGetMacro](#) ([PlanarConfiguration](#), int)
- [vtkGetMacro](#) ([Scale](#), double)
- [vtkGetMacro](#) ([Shift](#), double)
- [vtkGetObjectMacro](#) ([Curve](#), vtkPolyData)
- [vtkGetObjectMacro](#) ([DirectionCosines](#), vtkMatrix4x4)
- [vtkGetVector3Macro](#) ([ImagePositionPatient](#), double)
- [vtkGetVector6Macro](#) ([ImageOrientationPatient](#), double)
- [vtkSetMacro](#) ([ApplyLookupTable](#), int)
- [vtkSetMacro](#) ([LoadIconImage](#), int)
- [vtkSetMacro](#) ([LoadOverlays](#), int)
- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkTypeMacro](#) ([vtkGDCMImageReader2](#), vtkMedicalImageReader2)

Static Public Member Functions

- static [vtkGDCMImageReader2](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMImageReader2](#) ()
- [~vtkGDCMImageReader2](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char *filename, char *pointer, unsigned long &outlen)
- int [ProcessRequest](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- int [RequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- int [RequestDataCompat](#) ()
- int [RequestInformation](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char *)
- void [SetFilePrefix](#) (const char *)
- [vtkGetStringMacro](#) (FilePattern)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkSetVector6Macro](#) ([ImageOrientationPatient](#), double)

Protected Attributes

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRTToRGB](#)
- vtkPolyData * [Curve](#)
- vtkMatrix4x4 * [DirectionCosines](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]
- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

12.385.1 Detailed Description

Examples

[Compute3DSpacing.cxx](#).

12.385.2 Constructor & Destructor Documentation

12.385.2.1 [vtkGDCMImageReader2\(\)](#)

`vtkGDCMImageReader2::vtkGDCMImageReader2 ()` [protected]

References [vtkGDCMImageReader2\(\)](#).

Referenced by [vtkGDCMImageReader2\(\)](#), [~vtkGDCMImageReader2\(\)](#), [New\(\)](#), [vtkGetStringMacro\(\)](#), and [vtkTypeMacro\(\)](#).

12.385.2.2 [~vtkGDCMImageReader2\(\)](#)

`vtkGDCMImageReader2::~~vtkGDCMImageReader2 ()` [protected]

References [vtkGDCMImageReader2\(\)](#).

12.385.3 Member Function Documentation

12.385.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (  
    const char * fname) [virtual]
```

12.385.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (  
    const gdcm::ImageReader & reader) [protected]
```

References [FillMedicalImageInformation\(\)](#).

Referenced by [FillMedicalImageInformation\(\)](#).

12.385.3.3 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader2::GetDescriptiveName () [inline], [virtual]
```

12.385.3.4 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader2::GetFileExtensions () [inline], [virtual]
```

12.385.3.5 GetIconImage()

```
vtkImageData * vtkGDCMImageReader2::GetIconImage ()
```

12.385.3.6 GetIconImagePort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetIconImagePort ()
```

12.385.3.7 GetOverlay()

```
vtkImageData * vtkGDCMImageReader2::GetOverlay (  
    int i)
```

12.385.3.8 GetOverlayPort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetOverlayPort (  
    int index)
```

12.385.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (  
    const char * filename,  
    char * pointer,  
    unsigned long & outlen) [protected]
```

References [LoadSingleFile\(\)](#).

Referenced by [LoadSingleFile\(\)](#).

12.385.3.10 New()

```
vtkGDCMImageReader2 * vtkGDCMImageReader2::New () [static]
```

Examples

[Compute3DSpacing.cxx](#).

References [vtkGDCMImageReader2\(\)](#).

12.385.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (  
    ostream & os,  
    vtkIndent indent) [virtual]
```

12.385.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (  
    vtkInformation * request,  
    vtkInformationVector ** inputVector,  
    vtkInformationVector * outputVector) [protected]
```

References [ProcessRequest\(\)](#).

Referenced by [ProcessRequest\(\)](#).

12.385.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (  
    vtkInformation * request,  
    vtkInformationVector ** inputVector,  
    vtkInformationVector * outputVector) [protected]
```

References [RequestData\(\)](#).

Referenced by [RequestData\(\)](#).

12.385.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat () [protected]
```

References [RequestDataCompat\(\)](#).

Referenced by [RequestDataCompat\(\)](#).

12.385.3.15 RequestInformation()

```
int vtkGDCMImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector) [protected]
```

References [RequestInformation\(\)](#).

Referenced by [RequestInformation\(\)](#).

12.385.3.16 RequestInformationCompat()

```
int vtkGDCMImageReader2::RequestInformationCompat () [protected]
```

References [RequestInformationCompat\(\)](#).

Referenced by [RequestInformationCompat\(\)](#).

12.385.3.17 SetCurve()

```
virtual void vtkGDCMImageReader2::SetCurve (
    vtkPolyData * pd) [virtual]
```

References [SetCurve\(\)](#).

Referenced by [SetCurve\(\)](#).

12.385.3.18 SetFilePattern()

```
void vtkGDCMImageReader2::SetFilePattern (
    const char * ) [inline], [protected]
```

12.385.3.19 SetFilePrefix()

```
void vtkGDCMImageReader2::SetFilePrefix (
    const char * ) [inline], [protected]
```

References [SetFilePrefix\(\)](#).

Referenced by [SetFilePrefix\(\)](#).

12.385.3.20 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (  
    vtkMedicalImageProperties * pd) [virtual]
```

12.385.3.21 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (  
    ApplyLookupTable ,  
    int )
```

References [ApplyLookupTable](#).

12.385.3.22 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (  
    ApplyYBRToRGB ,  
    int )
```

References [ApplyYBRToRGB](#), and [vtkBooleanMacro\(\)](#).

12.385.3.23 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (  
    LoadIconImage ,  
    int )
```

References [LoadIconImage](#).

12.385.3.24 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (  
    LoadOverlays ,  
    int )
```

References [LoadOverlays](#).

Referenced by [vtkBooleanMacro\(\)](#).

12.385.3.25 vtkBooleanMacro() [5/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (  
    LossyFlag ,  
    int )
```

References [LossyFlag](#).

12.385.3.26 vtkGetMacro() [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

References [ApplyLookupTable](#).

12.385.3.27 vtkGetMacro() [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

References [ApplyYBRToRGB](#), and [vtkSetMacro\(\)](#).

12.385.3.28 vtkGetMacro() [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ImageFormat ,
    int )
```

References [ImageFormat](#), and [vtkGetMacro\(\)](#).

12.385.3.29 vtkGetMacro() [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadIconImage ,
    int )
```

References [LoadIconImage](#).

12.385.3.30 vtkGetMacro() [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

References [LoadOverlays](#).

Referenced by [vtkGetMacro\(\)](#), [vtkGetMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkGetMacro\(\)](#).

12.385.3.31 vtkGetMacro() [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LossyFlag ,
    int )
```

References [LossyFlag](#).

12.385.3.32 `vtkGetMacro()` [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

References [NumberOfIconImages](#).

12.385.3.33 `vtkGetMacro()` [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

References [NumberOfOverlays](#).

12.385.3.34 `vtkGetMacro()` [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

References [PlanarConfiguration](#), and [vtkGetMacro\(\)](#).

12.385.3.35 `vtkGetMacro()` [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Scale ,
    double )
```

References [Scale](#), and [vtkGetMacro\(\)](#).

12.385.3.36 `vtkGetMacro()` [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Shift ,
    double )
```

References [Shift](#), and [vtkGetMacro\(\)](#).

12.385.3.37 `vtkGetObjectMacro()` [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

References [Curve](#), and [vtkGetObjectMacro\(\)](#).

12.385.3.38 vtkGetObjectMacro() [2/2]

vtkGDCMImageReader2::vtkGetObjectMacro (
 [DirectionCosines](#) ,
 vtkMatrix4x4)

References [DirectionCosines](#).

Referenced by [vtkGetObjectMacro\(\)](#).

12.385.3.39 vtkGetStringMacro() [1/2]

vtkGDCMImageReader2::vtkGetStringMacro (
 FilePattern) [protected]

References [vtkGDCMImageReader2\(\)](#).

12.385.3.40 vtkGetStringMacro() [2/2]

vtkGDCMImageReader2::vtkGetStringMacro (
 FilePrefix) [protected]

12.385.3.41 vtkGetVector3Macro()

vtkGDCMImageReader2::vtkGetVector3Macro (
 [ImagePositionPatient](#) ,
 double)

References [ImagePositionPatient](#), and [vtkGetVector3Macro\(\)](#).

Referenced by [vtkGetVector3Macro\(\)](#).

12.385.3.42 vtkGetVector6Macro()

vtkGDCMImageReader2::vtkGetVector6Macro (
 [ImageOrientationPatient](#) ,
 double)

References [ImageOrientationPatient](#), and [vtkGetVector6Macro\(\)](#).

Referenced by [vtkGetVector6Macro\(\)](#).

12.385.3.43 vtkSetMacro() [1/4]

vtkGDCMImageReader2::vtkSetMacro (
 [ApplyLookupTable](#) ,
 int)

References [ApplyLookupTable](#).

12.385.3.44 `vtkSetMacro()` [2/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadIconImage ,
    int )
```

References [LoadIconImage](#).

12.385.3.45 `vtkSetMacro()` [3/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

References [LoadOverlays](#).

Referenced by [vtkGetMacro\(\)](#).

12.385.3.46 `vtkSetMacro()` [4/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LossyFlag ,
    int )
```

References [LossyFlag](#).

12.385.3.47 `vtkSetVector6Macro()`

```
vtkGDCMImageReader2::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

References [ImageOrientationPatient](#), and [vtkSetVector6Macro\(\)](#).

Referenced by [vtkSetVector6Macro\(\)](#).

12.385.3.48 `vtkTypeMacro()`

```
vtkGDCMImageReader2::vtkTypeMacro (
    vtkGDCMImageReader2 ,
    vtkMedicalImageReader2 )
```

References [vtkGDCMImageReader2\(\)](#).

12.385.4 Member Data Documentation

12.385.4.1 ApplyInverseVideo

int vtkGDCMImageReader2::ApplyInverseVideo [protected]

12.385.4.2 ApplyLookupTable

int vtkGDCMImageReader2::ApplyLookupTable [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.385.4.3 ApplyPlanarConfiguration

int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]

12.385.4.4 ApplyShiftScale

int vtkGDCMImageReader2::ApplyShiftScale [protected]

12.385.4.5 ApplyYBRToRGB

int vtkGDCMImageReader2::ApplyYBRToRGB [protected]

Referenced by [vtkBooleanMacro\(\)](#), and [vtkGetMacro\(\)](#).

12.385.4.6 Curve

vtkPolyData* vtkGDCMImageReader2::Curve [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.385.4.7 DirectionCosines

vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.385.4.8 ForceRescale

int vtkGDCMImageReader2::ForceRescale [protected]

12.385.4.9 IconDataScalarType

int vtkGDCMImageReader2::IconDataScalarType [protected]

12.385.4.10 IconImageDataExtent

int vtkGDCMImageReader2::IconImageDataExtent[6] [protected]

12.385.4.11 IconNumberOfScalarComponents

int vtkGDCMImageReader2::IconNumberOfScalarComponents [protected]

12.385.4.12 ImageFormat

int vtkGDCMImageReader2::ImageFormat [protected]

Referenced by [vtkGetMacro\(\)](#).

12.385.4.13 ImageOrientationPatient

double vtkGDCMImageReader2::ImageOrientationPatient[6] [protected]

Referenced by [vtkGetVector6Macro\(\)](#), and [vtkSetVector6Macro\(\)](#).

12.385.4.14 ImagePositionPatient

double vtkGDCMImageReader2::ImagePositionPatient[3] [protected]

Referenced by [vtkGetVector3Macro\(\)](#).

12.385.4.15 LoadIconImage

int vtkGDCMImageReader2::LoadIconImage [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.385.4.16 LoadOverlays

int vtkGDCMImageReader2::LoadOverlays [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.385.4.17 LossyFlag

int vtkGDCMImageReader2::LossyFlag [protected]

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.385.4.18 NumberOfIconImages

int vtkGDCMImageReader2::NumberOfIconImages [protected]

Referenced by [vtkGetMacro\(\)](#).

12.385.4.19 NumberOfOverlays

int vtkGDCMImageReader2::NumberOfOverlays [protected]

Referenced by [vtkGetMacro\(\)](#).

12.385.4.20 PlanarConfiguration

int vtkGDCMImageReader2::PlanarConfiguration [protected]

Referenced by [vtkGetMacro\(\)](#).

12.385.4.21 Scale

double vtkGDCMImageReader2::Scale [protected]

Referenced by [vtkGetMacro\(\)](#).

12.385.4.22 Shift

double vtkGDCMImageReader2::Shift [protected]

Referenced by [vtkGetMacro\(\)](#).

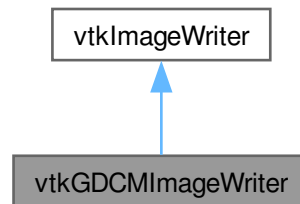
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

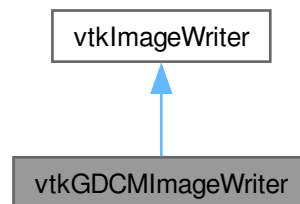
12.386 vtkGDCMImageWriter Class Reference

```
#include <vtkGDCMImageWriter.h>
```

Inheritance diagram for vtkGDCMImageWriter:



Collaboration diagram for vtkGDCMImageWriter:



Public Types

- `enum CompressionTypes` {
 `NO_COMPRESSION` = 0 ,
 `JPEG_COMPRESSION` ,
 `JPEG2000_COMPRESSION` ,
 `JPEGLS_COMPRESSION` ,
 `RLE_COMPRESSION` }

Public Member Functions

- virtual const char * [GetDescriptiveName](#) ()
- virtual const char * [GetFileExtensions](#) ()
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetDirectionCosines](#) (vtkMatrix4x4 *matrix)
- virtual void [SetDirectionCosinesFromImageOrientationPatient](#) (const double dircos[6])
- virtual void [SetFileNames](#) (vtkStringArray *)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties *)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (CompressionType, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetStringMacro](#) (SeriesUID)
- [vtkGetStringMacro](#) (StudyUID)
- [vtkSetMacro](#) (CompressionType, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (ImageFormat, int)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkSetMacro](#) (PlanarConfiguration, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetStringMacro](#) (SeriesUID)
- [vtkSetStringMacro](#) (StudyUID)
- [vtkTypeMacro](#) ([vtkGDCMImageWriter](#), vtkImageWriter)
- virtual void [Write](#) ()

Static Public Member Functions

- static [vtkGDCMImageWriter](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMImageWriter](#) ()
- [~vtkGDCMImageWriter](#) ()
- virtual char * [GetFileName](#) ()
- int [WriteGDCMData](#) (vtkImageData *data, int timeStep)
- void [WriteSlice](#) (vtkImageData *data)

12.386.1 Detailed Description

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), [RefCounting.cs](#), and [gdcmmorthoplanes.cxx](#).

12.386.2 Member Enumeration Documentation

12.386.2.1 CompressionTypes

enum [vtkGDCMImageWriter::CompressionTypes](#)

Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

12.386.3 Constructor & Destructor Documentation

12.386.3.1 vtkGDCMImageWriter()

[vtkGDCMImageWriter::vtkGDCMImageWriter\(\)](#) [protected]

Referenced by [GetFileName\(\)](#), [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.386.3.2 ~vtkGDCMImageWriter()

[vtkGDCMImageWriter::~~vtkGDCMImageWriter\(\)](#) [protected]

12.386.4 Member Function Documentation

12.386.4.1 GetDescriptiveName()

[virtual const char * vtkGDCMImageWriter::GetDescriptiveName\(\)](#) [inline], [virtual]

12.386.4.2 GetFileExtensions()

virtual const char * vtkGDCMImageWriter::GetFileExtensions () [inline], [virtual]

12.386.4.3 GetFileName()

virtual char * vtkGDCMImageWriter::GetFileName () [protected], [virtual]

References [vtkGDCMImageWriter\(\)](#).

12.386.4.4 New()

[vtkGDCMImageWriter](#) * vtkGDCMImageWriter::New () [static]

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), [RefCounting.cs](#), and [gdcmmorthoplanes.cxx](#).

References [vtkGDCMImageWriter\(\)](#).

12.386.4.5 PrintSelf()

virtual void vtkGDCMImageWriter::PrintSelf (
 ostream & os,
 vtkIndent indent) [virtual]

12.386.4.6 SetDirectionCosines()

virtual void vtkGDCMImageWriter::SetDirectionCosines (
 vtkMatrix4x4 * matrix) [virtual]

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), and [gdcmmorthoplanes.cxx](#).

12.386.4.7 SetDirectionCosinesFromImageOrientationPatient()

virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
 const double dircos[6]) [virtual]

12.386.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
    vtkStringArray * ) [virtual]
```

Examples

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

12.386.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * ) [virtual]
```

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), and [gdcmortoplanes.cxx](#).

12.386.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

12.386.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    LossyFlag ,
    int )
```

12.386.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    CompressionType ,
    int )
```

12.386.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    FileLowerLeft ,
    int )
```

12.386.4.14 vtkGetMacro() [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    ImageFormat ,
    int )
```

12.386.4.15 vtkGetMacro() [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    LossyFlag ,
    int )
```

12.386.4.16 vtkGetMacro() [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

12.386.4.17 vtkGetMacro() [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Scale ,
    double )
```

12.386.4.18 vtkGetMacro() [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Shift ,
    double )
```

12.386.4.19 vtkGetObjectMacro() [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

12.386.4.20 vtkGetObjectMacro() [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

12.386.4.21 vtkGetObjectMacro() [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

12.386.4.22 vtkGetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
    SeriesUID )
```

12.386.4.23 vtkGetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
    StudyUID )
```

12.386.4.24 vtkSetMacro() [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    CompressionType ,
    int )
```

12.386.4.25 vtkSetMacro() [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    FileLowerLeft ,
    int )
```

12.386.4.26 vtkSetMacro() [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    ImageFormat ,
    int )
```

12.386.4.27 vtkSetMacro() [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    LossyFlag ,
    int )
```

12.386.4.28 vtkSetMacro() [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    PlanarConfiguration ,
    int )
```

12.386.4.29 vtkSetMacro() [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    Scale ,
    double )
```

12.386.4.30 vtkSetMacro() [7/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    Shift ,
    double )
```

12.386.4.31 vtkSetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
    SeriesUID )
```

12.386.4.32 vtkSetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
    StudyUID )
```

12.386.4.33 vtkTypeMacro()

```
vtkGDCMImageWriter::vtkTypeMacro (
    vtkGDCMImageWriter ,
    vtkImageWriter )
```

References [vtkGDCMImageWriter\(\)](#).

12.386.4.34 Write()

```
virtual void vtkGDCMImageWriter::Write () [virtual]
```

Examples

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#),
[ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [HelloActiviz.cs](#),
[HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#),
and [gdcmmorthoplanes.cxx](#).

12.386.4.35 WriteGDCMData()

```
int vtkGDCMImageWriter::WriteGDCMData (  
    vtkImageData * data,  
    int timeStep) [protected]
```

12.386.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (  
    vtkImageData * data) [protected]
```

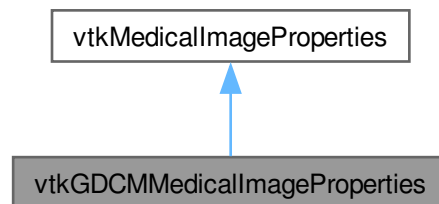
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

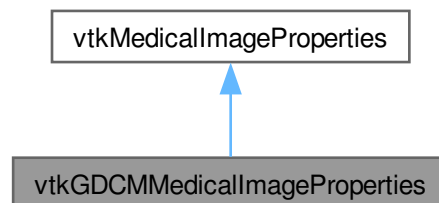
12.387 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void [Clear](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkGDCMMedicalImageProperties](#), vtkMedicalImageProperties)

Static Public Member Functions

- static [vtkGDCMMedicalImageProperties * New](#) ()

Protected Member Functions

- [vtkGDCMMedicalImageProperties](#) ()
- [~vtkGDCMMedicalImageProperties](#) ()
- [gdcmm::File](#) const & [GetFile](#) (unsigned int t)
- void [PushBackFile](#) ([gdcmm::File](#) const &f)

Friends

- class [vtkGDCMImageReader](#)
- class [vtkGDCMImageReader2](#)
- class [vtkGDCMImageWriter](#)

12.387.1 Constructor & Destructor Documentation

12.387.1.1 [vtkGDCMMedicalImageProperties\(\)](#)

[vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties](#) () [protected]

Referenced by [GetFile\(\)](#), [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.387.1.2 [~vtkGDCMMedicalImageProperties\(\)](#)

[vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties](#) () [protected]

12.387.2 Member Function Documentation

12.387.2.1 [Clear\(\)](#)

virtual void [vtkGDCMMedicalImageProperties::Clear](#) () [virtual]

12.387.2.2 GetFile()

```
gdcmm::File const & vtkGDCMMedicalImageProperties::GetFile (
    unsigned int t) [protected]
```

References [vtkGDCMMedicalImageProperties\(\)](#).

12.387.2.3 New()

```
vtkGDCMMedicalImageProperties * vtkGDCMMedicalImageProperties::New () [static]
```

References [vtkGDCMMedicalImageProperties\(\)](#).

12.387.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.387.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
    gdcmm::File const & f) [protected]
```

12.387.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
    vtkGDCMMedicalImageProperties ,
    vtkMedicalImageProperties )
```

References [vtkGDCMMedicalImageProperties\(\)](#).

12.387.3 Friends And Related Symbol Documentation

12.387.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader [friend]
```

References [vtkGDCMImageReader](#).

Referenced by [vtkGDCMImageReader](#).

12.387.3.2 vtkGDCMImageReader2

friend class [vtkGDCMImageReader2](#) [friend]

References [vtkGDCMImageReader2](#).

Referenced by [vtkGDCMImageReader2](#).

12.387.3.3 vtkGDCMImageWriter

friend class [vtkGDCMImageWriter](#) [friend]

References [vtkGDCMImageWriter](#).

Referenced by [vtkGDCMImageWriter](#).

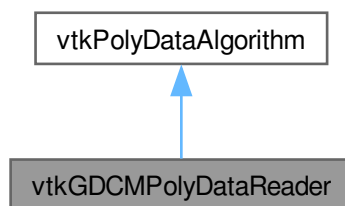
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

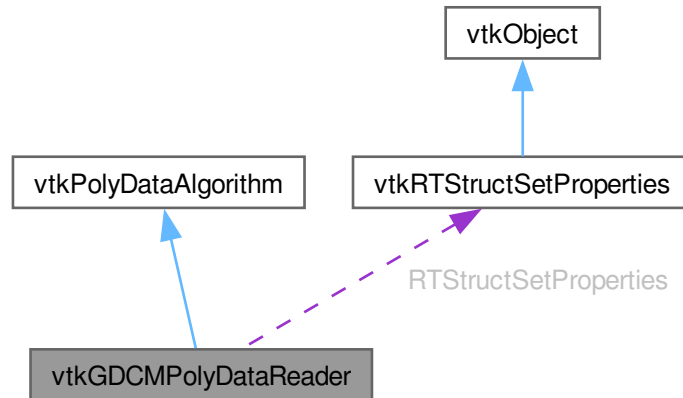
12.388 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

Inheritance diagram for vtkGDCMPolyDataReader:



Collaboration diagram for `vtkGDCMPolyDataReader`:



Public Member Functions

- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetObjectMacro](#) (RTStructSetProperties, vtkRTStructSetProperties)
- [vtkGetStringMacro](#) (FileName)
- [vtkSetStringMacro](#) (FileName)
- [vtkTypeMacro](#) (vtkGDCMPolyDataReader, vtkPolyDataAlgorithm)

Static Public Member Functions

- static [vtkGDCMPolyDataReader * New](#) ()

Protected Member Functions

- [vtkGDCMPolyDataReader](#) ()
- [~vtkGDCMPolyDataReader](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::Reader](#) &reader)
- int [RequestData](#) (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- int [RequestData_HemodynamicWaveformStorage](#) ([gdcm::Reader](#) const &reader, vtkInformationVector *outputVector)
- int [RequestData_RTStructureSetStorage](#) ([gdcm::Reader](#) const &reader, vtkInformationVector *outputVector)
- int [RequestInformation](#) (vtkInformation *vtkNotUsed(request), vtkInformationVector **vtkNotUsed(inputVector), vtkInformationVector *outputVector)
- int [RequestInformation_HemodynamicWaveformStorage](#) ([gdcm::Reader](#) const &reader)
- int [RequestInformation_RTStructureSetStorage](#) ([gdcm::Reader](#) const &reader)

Protected Attributes

- `char * FileName`
- `vtkMedicalImageProperties * MedicalImageProperties`
- `vtkRTStructSetProperties * RTStructSetProperties`

12.388.1 Detailed Description

Examples

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

12.388.2 Constructor & Destructor Documentation

12.388.2.1 `vtkGDCMPolyDataReader()`

`vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ()` [protected]

Referenced by [New\(\)](#), [RequestData__HemodynamicWaveformStorage\(\)](#), and [vtkTypeMacro\(\)](#).

12.388.2.2 `~vtkGDCMPolyDataReader()`

`vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ()` [protected]

12.388.3 Member Function Documentation

12.388.3.1 `FillMedicalImageInformation()`

`void vtkGDCMPolyDataReader::FillMedicalImageInformation (`
 `const gdcm::Reader & reader)` [protected]

12.388.3.2 `New()`

`vtkGDCMPolyDataReader * vtkGDCMPolyDataReader::New ()` [static]

Examples

[GenerateRTSTRUCT.cxx](#), [gdcmscene.cxx](#), and [rtstructapp.cxx](#).

References [vtkGDCMPolyDataReader\(\)](#).

12.388.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
    ostream & os,
    vtkIndent indent) [virtual]
```

12.388.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected]
```

12.388.3.5 RequestData_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
    gdcmm::Reader const & reader,
    vtkInformationVector * outputVector) [protected]
```

References [vtkGDCMPolyDataReader\(\)](#).

12.388.3.6 RequestData_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
    gdcmm::Reader const & reader,
    vtkInformationVector * outputVector) [protected]
```

12.388.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
    vtkInformation * vtkNotUsedrequest,
    vtkInformationVector ** vtkNotUsedinputVector,
    vtkInformationVector * outputVector) [protected]
```

12.388.3.8 RequestInformation_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
    gdcmm::Reader const & reader) [protected]
```

12.388.3.9 RequestInformation_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
    gdcmm::Reader const & reader) [protected]
```

12.388.3.10 vtkGetObjectMacro() [1/2]

vtkGDCMPolyDataReader::vtkGetObjectMacro (
 [MedicalImageProperties](#) ,
 vtkMedicalImageProperties)

References [MedicalImageProperties](#).

12.388.3.11 vtkGetObjectMacro() [2/2]

vtkGDCMPolyDataReader::vtkGetObjectMacro (
 [RTStructSetProperties](#) ,
 vtkRTStructSetProperties)

References [RTStructSetProperties](#).

12.388.3.12 vtkGetStringMacro()

vtkGDCMPolyDataReader::vtkGetStringMacro (
 [FileName](#))

References [FileName](#).

12.388.3.13 vtkSetStringMacro()

vtkGDCMPolyDataReader::vtkSetStringMacro (
 [FileName](#))

References [FileName](#).

12.388.3.14 vtkTypeMacro()

vtkGDCMPolyDataReader::vtkTypeMacro (
 [vtkGDCMPolyDataReader](#) ,
 vtkPolyDataAlgorithm)

References [vtkGDCMPolyDataReader\(\)](#).

12.388.4 Member Data Documentation

12.388.4.1 FileName

char* vtkGDCMPolyDataReader::FileName [protected]

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.388.4.2 MedicalImageProperties

`vtkMedicalImageProperties*` `vtkGDCMPolyDataReader::MedicalImageProperties` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.388.4.3 RTStructSetProperties

[vtkRTStructSetProperties*](#) `vtkGDCMPolyDataReader::RTStructSetProperties` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

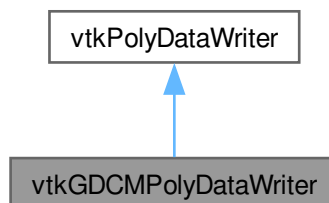
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

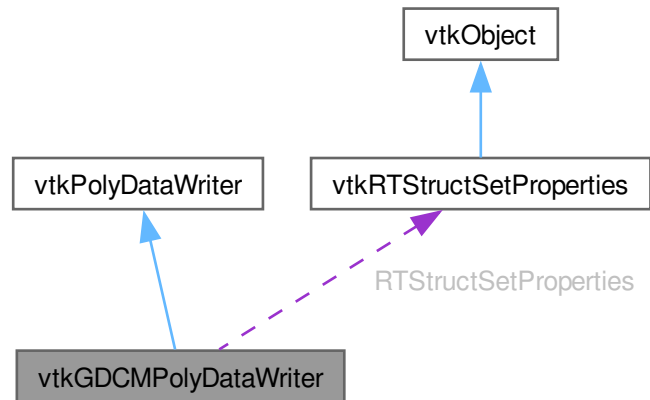
12.389 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for `vtkGDCMPolyDataWriter`:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void [InitializeRTStructSet](#) (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray *inROINames, vtkStringArray *inROIAlgorithmName, vtkStringArray *inROIType)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties *pd)
- void [SetNumberOfInputPorts](#) (int n)
- virtual void [SetRTStructSetProperties](#) (vtkRTStructSetProperties *pd)
- [vtkTypeMacro](#) (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static [vtkGDCMPolyDataWriter * New](#) ()

Protected Member Functions

- [vtkGDCMPolyDataWriter](#) ()
- [~vtkGDCMPolyDataWriter](#) ()
- void [WriteData](#) ()
- void [WriteRTSTRUCTData](#) (gdcmm::File &file, int num)
- void [WriteRTSTRUCTInfo](#) (gdcmm::File &file)

Protected Attributes

- vtkMedicalImageProperties * [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) * [RTStructSetProperties](#)

12.389.1 Detailed Description

Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

12.389.2 Constructor & Destructor Documentation

12.389.2.1 vtkGDCMPolyDataWriter()

`vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ()` [protected]

Referenced by [New\(\)](#), [vtkTypeMacro\(\)](#), and [WriteRTSTRUCTData\(\)](#).

12.389.2.2 ~vtkGDCMPolyDataWriter()

`vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ()` [protected]

12.389.3 Member Function Documentation

12.389.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
    vtkStdString inDirectory,
    vtkStdString inStructLabel,
    vtkStdString inStructName,
    vtkStringArray * inROINames,
    vtkStringArray * inROIAlgorithmName,
    vtkStringArray * inROIType)
```

Examples

[GenerateRTSTRUCT.cxx](#).

12.389.3.2 New()

`vtkGDCMPolyDataWriter * vtkGDCMPolyDataWriter::New ()` [static]

Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

References [vtkGDCMPolyDataWriter\(\)](#).

12.389.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
    ostream & os,
    vtkIndent indent) [virtual]
```

12.389.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd) [virtual]
```

Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

12.389.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
    int n)
```

Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

12.389.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
    vtkRTStructSetProperties * pd) [virtual]
```

Examples

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

12.389.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
    vtkGDCMPolyDataWriter ,
    vtkPolyDataWriter )
```

References [vtkGDCMPolyDataWriter\(\)](#).

12.389.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData () [protected]
```

12.389.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
    gdc::File & file,
    int num) [protected]
```

References [vtkGDCMPolyDataWriter\(\)](#).

12.389.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
    gdc::File & file) [protected]
```

12.389.4 Member Data Documentation

12.389.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]
```

12.389.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]
```

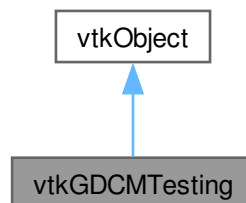
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

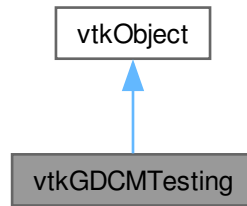
12.390 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



Public Types

- typedef const char *const (* [MD5MetaImagesType](#))[3]

Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkGDCMTesting](#), vtkObject)

Static Public Member Functions

- static const char * [GetGDCMDataRoot](#) ()
- static const char *const * [GetMD5MetaImage](#) (unsigned int file)
- static const char * [GetMHDMD5FromFile](#) (const char *filepath)
- static unsigned int [GetNumberOfMD5MetaImages](#) ()
- static const char * [GetRAWMD5FromFile](#) (const char *filepath)
- static const char * [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

12.390.1 Detailed Description

Examples

[HelloActiviz5.cs](#), [HelloVTKWorld2.cs](#), [MetaImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

12.390.2 Member Typedef Documentation

12.390.2.1 MD5MetaImagesType

```
typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]
```

12.390.3 Constructor & Destructor Documentation

12.390.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting () [protected]
```

Referenced by [~vtkGDCMTesting\(\)](#), [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.390.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~~vtkGDCMTesting () [protected]
```

References [vtkGDCMTesting\(\)](#).

12.390.4 Member Function Documentation

12.390.4.1 GetGDCMDataRoot()

```
const char * vtkGDCMTesting::GetGDCMDataRoot () [static]
```

Examples

[HelloActiviz5.cs](#), and [ReadSeriesIntoVTK.java](#).

12.390.4.2 GetMD5MetaImage()

```
const char *const * vtkGDCMTesting::GetMD5MetaImage (
    unsigned int file) [static]
```

12.390.4.3 GetMHDMD5FromFile()

```
const char * vtkGDCMTesting::GetMHDMD5FromFile (
    const char * filepath) [static]
```

Examples

[MetaImageMD5Activiz.cs](#).

12.390.4.4 GetNumberOfMD5MetaImages()

```
unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages () [static]
```

12.390.4.5 GetRAWMD5FromFile()

```
const char * vtkGDCMTesting::GetRAWMD5FromFile (  
    const char * filepath) [static]
```

Examples

[MetaImageMD5Activiz.cs](#).

12.390.4.6 GetVTKDataRoot()

```
const char * vtkGDCMTesting::GetVTKDataRoot () [static]
```

Examples

[HelloActiviz5.cs](#), and [HelloVTKWorld2.cs](#).

12.390.4.7 New()

```
vtkGDCMTesting * vtkGDCMTesting::New () [static]
```

Examples

[RefCounting.cs](#).

References [vtkGDCMTesting\(\)](#).

12.390.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (  
    ostream & os,  
    vtkIndent indent)
```

12.390.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (  
    vtkGDCMTesting ,  
    vtkObject )
```

References [vtkGDCMTesting\(\)](#).

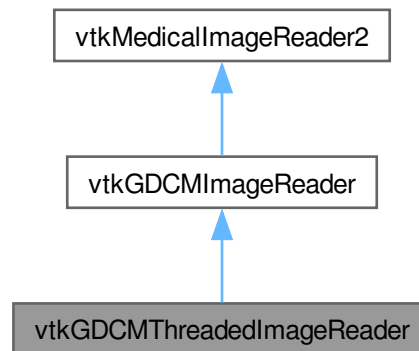
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

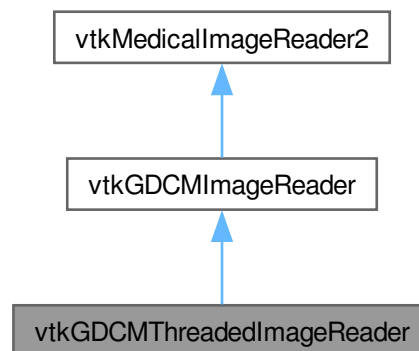
12.391 vtkGDCMThreadedImageReader Class Reference

```
#include <vtkGDCMThreadedImageReader.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader:



Collaboration diagram for vtkGDCMThreadedImageReader:



Public Member Functions

- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkBooleanMacro](#) (UseShiftScale, int)

- [vtkGetMacro](#) (UseShiftScale, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkTypeMacro](#) (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Public Member Functions inherited from [vtkGDCMImageReader](#)

- virtual int [CanReadFile](#) (const char *fname)
- virtual const char * [GetDescriptiveName](#) ()
- virtual const char * [GetFileExtensions](#) ()
- vtkImageData * [GetIconImage](#) ()
- vtkImageData * [GetOverlay](#) (int i)
- virtual void [SetCurve](#) (vtkPolyData *pd)
- virtual void [SetFileNames](#) (vtkStringArray *)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties *pd)
- [vtkBooleanMacro](#) (ApplyLookupTable, int)
- int [vtkBooleanMacro](#) (ApplyYBRToRGB, int)
- [vtkBooleanMacro](#) (LoadIconImage, int)
- [vtkBooleanMacro](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (ApplyLookupTable, int)
- [vtkGetMacro](#) (ApplyYBRToRGB, int) [vtkSetMacro](#)(ApplyYBRToRGB
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (LoadIconImage, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (NumberOfIconImages, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetObjectMacro](#) (Curve, vtkPolyData)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetVector3Macro](#) (ImagePositionPatient, double)
- [vtkGetVector6Macro](#) (ImageOrientationPatient, double)
- [vtkSetMacro](#) (ApplyLookupTable, int)
- [vtkSetMacro](#) (LoadIconImage, int)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkTypeMacro](#) (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static [vtkGDCMThreadedImageReader](#) * [New](#) ()

Static Public Member Functions inherited from [vtkGDCMImageReader](#)

- static [vtkGDCMImageReader](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMThreadedImageReader](#) ()
- [~vtkGDCMThreadedImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject *out)
- void [ExecuteInformation](#) ()
- void [ReadFiles](#) (unsigned int nfiles, const char *filenames[])
- void [RequestDataCompat](#) ()

Protected Member Functions inherited from [vtkGDCMImageReader](#)

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject *out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char *filename, char *pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char *)
- void [SetFilePrefix](#) (const char *)
- [vtkGetStringMacro](#) (FilePattern)
- [vtkGetStringMacro](#) (FilePrefix)
- [vtkSetVector6Macro](#) ([ImageOrientationPatient](#), double)

Additional Inherited Members

Protected Attributes inherited from [vtkGDCMImageReader](#)

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- vtkPolyData * [Curve](#)
- vtkMatrix4x4 * [DirectionCosines](#)
- vtkStringArray * [FileNames](#)
- int [ForceRescale](#)
- int [IconDataScalarType](#)
- int [IconImageDataExtent](#) [6]
- int [IconNumberOfScalarComponents](#)
- int [ImageFormat](#)
- double [ImageOrientationPatient](#) [6]
- double [ImagePositionPatient](#) [3]

- int [LoadIconImage](#)
- int [LoadOverlays](#)
- int [LossyFlag](#)
- vtkMedicalImageProperties * [MedicalImageProperties](#)
- int [NumberOfIconImages](#)
- int [NumberOfOverlays](#)
- int [PlanarConfiguration](#)
- double [Scale](#)
- double [Shift](#)

12.391.1 Constructor & Destructor Documentation

12.391.1.1 vtkGDCMThreadedImageReader()

`vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ()` [protected]

Referenced by [New\(\)](#), [RequestDataCompat\(\)](#), and [vtkTypeMacro\(\)](#).

12.391.1.2 ~vtkGDCMThreadedImageReader()

`vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ()` [protected]

12.391.2 Member Function Documentation

12.391.2.1 ExecuteData()

`void vtkGDCMThreadedImageReader::ExecuteData (`
`vtkDataObject * out)` [protected]

12.391.2.2 ExecuteInformation()

`void vtkGDCMThreadedImageReader::ExecuteInformation ()` [protected]

12.391.2.3 New()

`vtkGDCMThreadedImageReader * vtkGDCMThreadedImageReader::New ()` [static]

References [vtkGDCMThreadedImageReader\(\)](#).

12.391.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (  
    ostream & os,  
    vtkIndent indent) [virtual]
```

Reimplemented from [vtkGDCMImageReader](#).

12.391.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (  
    unsigned int nfiles,  
    const char * filenames[]) [protected]
```

12.391.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat () [protected]
```

References [vtkGDCMThreadedImageReader\(\)](#).

12.391.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (  
    UseShiftScale ,  
    int )
```

12.391.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (  
    UseShiftScale ,  
    int )
```

12.391.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (  
    Scale ,  
    double )
```

References [vtkGDCMImageReader::Scale](#).

12.391.2.10 vtkSetMacro() [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (  
    Shift ,  
    double )
```

References [vtkGDCMImageReader::Shift](#).

12.391.2.11 vtkSetMacro() [3/3]

```

vtkGDCMThreadedImageReader::vtkSetMacro (
    UseShiftScale ,
    int )

```

12.391.2.12 vtkTypeMacro()

```

vtkGDCMThreadedImageReader::vtkTypeMacro (
    vtkGDCMThreadedImageReader ,
    vtkGDCMImageReader )

```

References [vtkGDCMImageReader::vtkGDCMImageReader\(\)](#), and [vtkGDCMThreadedImageReader\(\)](#).

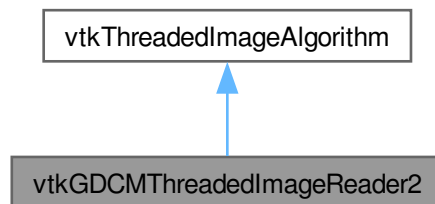
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

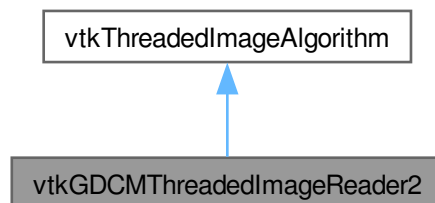
12.392 vtkGDCMThreadedImageReader2 Class Reference

```
#include <vtkGDCMThreadedImageReader2.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader2:



Collaboration diagram for vtkGDCMThreadedImageReader2:



Public Member Functions

- virtual const char * [GetFileName](#) (int i=0)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetFileName](#) (const char *filename)
- virtual void [SetFileNames](#) (vtkStringArray *)
- int [SplitExtent](#) (int splitExt[6], int startExt[6], int num, int total)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkBooleanMacro](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (UseShiftScale, int)
- [vtkGetMacro](#) (DataScalarType, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (NumberOfScalarComponents, int)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (UseShiftScale, int)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetVector3Macro](#) (DataOrigin, double)
- [vtkGetVector3Macro](#) (DataSpacing, double)
- [vtkGetVector6Macro](#) (DataExtent, int)
- [vtkSetMacro](#) (DataScalarType, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (NumberOfScalarComponents, int)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkSetVector3Macro](#) (DataOrigin, double)
- [vtkSetVector3Macro](#) (DataSpacing, double)
- [vtkSetVector6Macro](#) (DataExtent, int)
- [vtkTypeMacro](#) ([vtkGDCMThreadedImageReader2](#), vtkThreadedImageAlgorithm)

Static Public Member Functions

- static [vtkGDCMThreadedImageReader2](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMThreadedImageReader2](#) ()
- [~vtkGDCMThreadedImageReader2](#) ()
- int [RequestInformation](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- void [ThreadedRequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int outExt[6], int id)

12.392.1 Constructor & Destructor Documentation

12.392.1.1 vtkGDCMThreadedImageReader2()

vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 () [protected]

Referenced by [New\(\)](#), [ThreadedRequestData\(\)](#), and [vtkTypeMacro\(\)](#).

12.392.1.2 ~vtkGDCMThreadedImageReader2()

vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2 () [protected]

12.392.2 Member Function Documentation

12.392.2.1 GetFileName()

virtual const char * vtkGDCMThreadedImageReader2::GetFileName (
int i = 0) [virtual]

12.392.2.2 New()

[vtkGDCMThreadedImageReader2](#) * vtkGDCMThreadedImageReader2::New () [static]

References [vtkGDCMThreadedImageReader2\(\)](#).

12.392.2.3 PrintSelf()

virtual void vtkGDCMThreadedImageReader2::PrintSelf (
ostream & os,
vtkIndent indent) [virtual]

12.392.2.4 RequestInformation()

int vtkGDCMThreadedImageReader2::RequestInformation (
vtkInformation * request,
vtkInformationVector ** inputVector,
vtkInformationVector * outputVector) [protected]

12.392.2.5 SetFileName()

virtual void vtkGDCMThreadedImageReader2::SetFileName (
const char * filename) [virtual]

12.392.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
    vtkStringArray * ) [virtual]
```

12.392.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
    int splitExt[6],
    int startExt[6],
    int num,
    int total)
```

12.392.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int outExt[6],
    int id) [protected]
```

References [vtkGDCMThreadedImageReader2\(\)](#).

12.392.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

12.392.2.10 vtkBooleanMacro() [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

12.392.2.11 vtkBooleanMacro() [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    UseShiftScale ,
    int )
```


12.392.2.12 vtkGetMacro() [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    DataScalarType ,
    int )
```

12.392.2.13 vtkGetMacro() [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    FileLowerLeft ,
    int )
```

12.392.2.14 vtkGetMacro() [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

12.392.2.15 vtkGetMacro() [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

12.392.2.16 vtkGetMacro() [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfScalarComponents ,
    int )
```

12.392.2.17 vtkGetMacro() [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Scale ,
    double )
```

12.392.2.18 vtkGetMacro() [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Shift ,
    double )
```

12.392.2.19 vtkGetMacro() [8/8]

vtkGDCMThreadedImageReader2::vtkGetMacro (
 UseShiftScale ,
 int)

12.392.2.20 vtkGetObjectMacro()

vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
 FileNames ,
 vtkStringArray)

12.392.2.21 vtkGetVector3Macro() [1/2]

vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
 DataOrigin ,
 double)

12.392.2.22 vtkGetVector3Macro() [2/2]

vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
 DataSpacing ,
 double)

12.392.2.23 vtkGetVector6Macro()

vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
 DataExtent ,
 int)

12.392.2.24 vtkSetMacro() [1/7]

vtkGDCMThreadedImageReader2::vtkSetMacro (
 DataScalarType ,
 int)

12.392.2.25 vtkSetMacro() [2/7]

vtkGDCMThreadedImageReader2::vtkSetMacro (
 FileLowerLeft ,
 int)

12.392.2.26 vtkSetMacro() [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

12.392.2.27 vtkSetMacro() [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    NumberOfScalarComponents ,
    int )
```

12.392.2.28 vtkSetMacro() [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Scale ,
    double )
```

12.392.2.29 vtkSetMacro() [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Shift ,
    double )
```

12.392.2.30 vtkSetMacro() [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    UseShiftScale ,
    int )
```

12.392.2.31 vtkSetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataOrigin ,
    double )
```

12.392.2.32 vtkSetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataSpacing ,
    double )
```

12.392.2.33 vtkSetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
    DataExtent ,
    int )
```

12.392.2.34 vtkTypeMacro()

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
    vtkGDCMThreadedImageReader2 ,
    vtkThreadedImageAlgorithm )
```

References [vtkGDCMThreadedImageReader2\(\)](#).

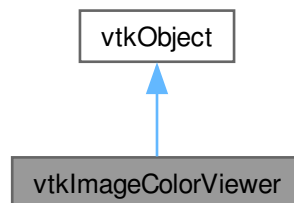
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

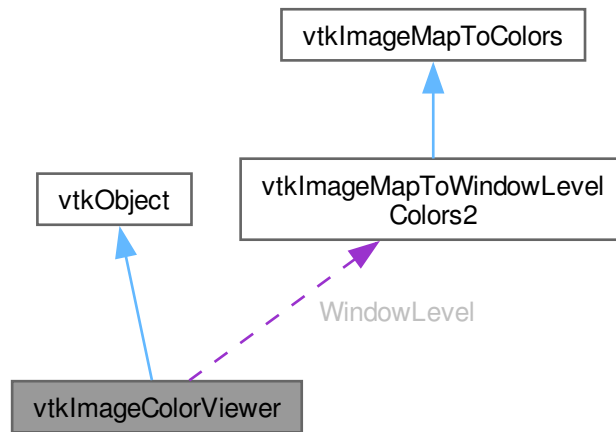
12.393 vtkImageColorViewer Class Reference

```
#include <vtkImageColorViewer.h>
```

Inheritance diagram for vtkImageColorViewer:



Collaboration diagram for vtkImageColorViewer:



Public Types

- enum {
[SLICE_ORIENTATION_YZ](#) = 0 ,
[SLICE_ORIENTATION_XZ](#) = 1 ,
[SLICE_ORIENTATION_XY](#) = 2 }

Public Member Functions

- virtual void [AddInput](#) (vtkImageData *input)
- virtual void [AddInputConnection](#) (vtkAlgorithmOutput *input)
- virtual double [GetColorLevel](#) ()
- virtual double [GetColorWindow](#) ()
- virtual vtkImageData * [GetInput](#) ()
- virtual int [GetOffScreenRendering](#) ()
- double [GetOverlayVisibility](#) ()
- virtual int * [GetPosition](#) ()
- virtual int * [GetSize](#) ()
- virtual int [GetSliceMax](#) ()
- virtual int [GetSliceMin](#) ()
- virtual int * [GetSliceRange](#) ()
- virtual void [GetSliceRange](#) (int &min, int &max)
- virtual void [GetSliceRange](#) (int range[2])
- virtual const char * [GetWindowName](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [Render](#) (void)
- virtual void [SetColorLevel](#) (double s)

- virtual void [SetColorWindow](#) (double s)
- virtual void [SetDisplayId](#) (void *a)
- virtual void [SetInput](#) (vtkImageData *in)
- virtual void [SetInputConnection](#) (vtkAlgorithmOutput *input)
- virtual void [SetOffScreenRendering](#) (int)
- void [SetOverlayVisibility](#) (double vis)
- virtual void [SetParentId](#) (void *a)
- virtual void [SetPosition](#) (int a, int b)
- virtual void [SetPosition](#) (int a[2])
- virtual void [SetRenderer](#) (vtkRenderer *arg)
- virtual void [SetRenderWindow](#) (vtkRenderWindow *arg)
- virtual void [SetSize](#) (int a, int b)
- virtual void [SetSize](#) (int a[2])
- virtual void [SetSlice](#) (int s)
- virtual void [SetSliceOrientation](#) (int orientation)
- virtual void [SetSliceOrientationToXY](#) ()
- virtual void [SetSliceOrientationToXZ](#) ()
- virtual void [SetSliceOrientationToYZ](#) ()
- virtual void [SetupInteractor](#) (vtkRenderWindowInteractor *)
- virtual void [SetWindowId](#) (void *a)
- virtual void [UpdateDisplayExtent](#) ()
- [vtkBooleanMacro](#) (OffScreenRendering, int)
- [vtkGetMacro](#) (Slice, int)
- [vtkGetMacro](#) (SliceOrientation, int)
- [vtkGetObjectMacro](#) (ImageActor, vtkImageActor)
- [vtkGetObjectMacro](#) (InteractorStyle, vtkInteractorStyleImage)
- [vtkGetObjectMacro](#) (Renderer, vtkRenderer)
- [vtkGetObjectMacro](#) (RenderWindow, vtkRenderWindow)
- [vtkGetObjectMacro](#) (WindowLevel, vtkImageMapToWindowLevelColors2)
- [vtkTypeMacro](#) (vtkImageColorViewer, vtkObject)

Static Public Member Functions

- static [vtkImageColorViewer * New](#) ()

Protected Member Functions

- [vtkImageColorViewer](#) ()
- [~vtkImageColorViewer](#) ()
- virtual void [InstallPipeline](#) ()
- virtual void [UnInstallPipeline](#) ()
- virtual void [UpdateOrientation](#) ()

Protected Attributes

- int [FirstRender](#)
- vtkImageActor * [ImageActor](#)
- vtkRenderWindowInteractor * [Interactor](#)
- vtkInteractorStyleImage * [InteractorStyle](#)
- vtkImageActor * [OverlayImageActor](#)
- vtkRenderer * [Renderer](#)
- vtkRenderWindow * [RenderWindow](#)
- int [Slice](#)
- int [SliceOrientation](#)
- [vtkImageMapToWindowLevelColors2](#) * [WindowLevel](#)

Friends

- class [vtkImageColorViewerCallback](#)

12.393.1 Detailed Description

Examples

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

12.393.2 Member Enumeration Documentation

12.393.2.1 anonymous enum

anonymous enum

Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

12.393.3 Constructor & Destructor Documentation

12.393.3.1 vtkImageColorViewer()

`vtkImageColorViewer::vtkImageColorViewer ()` [protected]

Referenced by [New\(\)](#), [vtkImageColorViewerCallback](#), and [vtkTypeMacro\(\)](#).

12.393.3.2 ~vtkImageColorViewer()

vtkImageColorViewer::~~vtkImageColorViewer () [protected]

12.393.4 Member Function Documentation

12.393.4.1 AddInput()

virtual void vtkImageColorViewer::AddInput (
 vtkImageData * input) [virtual]

12.393.4.2 AddInputConnection()

virtual void vtkImageColorViewer::AddInputConnection (
 vtkAlgorithmOutput * input) [virtual]

12.393.4.3 GetColorLevel()

virtual double vtkImageColorViewer::GetColorLevel () [virtual]

12.393.4.4 GetColorWindow()

virtual double vtkImageColorViewer::GetColorWindow () [virtual]

12.393.4.5 GetInput()

virtual vtkImageData * vtkImageColorViewer::GetInput () [virtual]

12.393.4.6 GetOffScreenRendering()

virtual int vtkImageColorViewer::GetOffScreenRendering () [virtual]

12.393.4.7 GetOverlayVisibility()

double vtkImageColorViewer::GetOverlayVisibility ()

12.393.4.8 GetPosition()

virtual int * vtkImageColorViewer::GetPosition () [virtual]

12.393.4.9 GetSize()

virtual int * vtkImageColorViewer::GetSize () [virtual]

12.393.4.10 GetSliceMax()

virtual int vtkImageColorViewer::GetSliceMax () [virtual]

12.393.4.11 GetSliceMin()

virtual int vtkImageColorViewer::GetSliceMin () [virtual]

12.393.4.12 GetSliceRange() [1/3]

virtual int * vtkImageColorViewer::GetSliceRange () [virtual]

12.393.4.13 GetSliceRange() [2/3]

virtual void vtkImageColorViewer::GetSliceRange (
 int & min,
 int & max) [virtual]

12.393.4.14 GetSliceRange() [3/3]

virtual void vtkImageColorViewer::GetSliceRange (
 int range[2]) [inline], [virtual]

References [GetSliceRange\(\)](#).

Referenced by [GetSliceRange\(\)](#).

12.393.4.15 GetWindowName()

virtual const char * vtkImageColorViewer::GetWindowName () [virtual]

12.393.4.16 InstallPipeline()

virtual void vtkImageColorViewer::InstallPipeline () [protected], [virtual]

12.393.4.17 New()

[vtkImageColorViewer](#) * [vtkImageColorViewer::New](#) () [static]

Examples

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

References [vtkImageColorViewer\(\)](#).

12.393.4.18 PrintSelf()

```
void vtkImageColorViewer::PrintSelf (  
    ostream & os,  
    vtkIndent indent)
```

12.393.4.19 Render()

```
virtual void vtkImageColorViewer::Render (  
    void ) [virtual]
```

Examples

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

12.393.4.20 SetColorLevel()

```
virtual void vtkImageColorViewer::SetColorLevel (  
    double s) [virtual]
```

12.393.4.21 SetColorWindow()

```
virtual void vtkImageColorViewer::SetColorWindow (  
    double s) [virtual]
```

12.393.4.22 SetDisplayId()

```
virtual void vtkImageColorViewer::SetDisplayId (  
    void * a) [virtual]
```

12.393.4.23 SetInput()

```
virtual void vtkImageColorViewer::SetInput (  
    vtkImageData * in) [virtual]
```

Examples

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

12.393.4.24 SetInputConnection()

```
virtual void vtkImageColorViewer::SetInputConnection (  
    vtkAlgorithmOutput * input) [virtual]
```

12.393.4.25 SetOffScreenRendering()

```
virtual void vtkImageColorViewer::SetOffScreenRendering (  
    int ) [virtual]
```

12.393.4.26 SetOverlayVisibility()

```
void vtkImageColorViewer::SetOverlayVisibility (  
    double vis)
```

12.393.4.27 SetParentId()

```
virtual void vtkImageColorViewer::SetParentId (  
    void * a) [virtual]
```

12.393.4.28 SetPosition() [1/2]

```
virtual void vtkImageColorViewer::SetPosition (  
    int a,  
    int b) [virtual]
```

12.393.4.29 SetPosition() [2/2]

```
virtual void vtkImageColorViewer::SetPosition (  
    int a[2]) [inline], [virtual]
```

References [SetPosition\(\)](#).

Referenced by [SetPosition\(\)](#).

12.393.4.30 SetRenderer()

```
virtual void vtkImageColorViewer::SetRenderer (  
    vtkRenderer * arg) [virtual]
```

12.393.4.31 SetRenderWindow()

```
virtual void vtkImageColorViewer::SetRenderWindow (  
    vtkRenderWindow * arg) [virtual]
```

12.393.4.32 SetSize() [1/2]

```
virtual void vtkImageColorViewer::SetSize (  
    int a,  
    int b) [virtual]
```

Examples

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

12.393.4.33 SetSize() [2/2]

```
virtual void vtkImageColorViewer::SetSize (  
    int a[2]) [inline], [virtual]
```

References [SetSize\(\)](#).

Referenced by [SetSize\(\)](#).

12.393.4.34 SetSlice()

```
virtual void vtkImageColorViewer::SetSlice (  
    int s) [virtual]
```

12.393.4.35 SetSliceOrientation()

```
virtual void vtkImageColorViewer::SetSliceOrientation (  
    int orientation) [virtual]
```

Referenced by [SetSliceOrientationToXY\(\)](#), [SetSliceOrientationToXZ\(\)](#), and [SetSliceOrientationToYZ\(\)](#).

12.393.4.36 SetSliceOrientationToXY()

virtual void vtkImageColorViewer::SetSliceOrientationToXY () [inline], [virtual]

References [SetSliceOrientation\(\)](#), and [SLICE_ORIENTATION_XY](#).

12.393.4.37 SetSliceOrientationToXZ()

virtual void vtkImageColorViewer::SetSliceOrientationToXZ () [inline], [virtual]

References [SetSliceOrientation\(\)](#), and [SLICE_ORIENTATION_XZ](#).

12.393.4.38 SetSliceOrientationToYZ()

virtual void vtkImageColorViewer::SetSliceOrientationToYZ () [inline], [virtual]

References [SetSliceOrientation\(\)](#), and [SLICE_ORIENTATION_YZ](#).

12.393.4.39 SetupInteractor()

virtual void vtkImageColorViewer::SetupInteractor (
 vtkRenderWindowInteractor *) [virtual]

Examples

[gdcmrtonplan.cxx](#), and [gdcmrtpplan.cxx](#).

12.393.4.40 SetWindowId()

virtual void vtkImageColorViewer::SetWindowId (
 void * a) [virtual]

12.393.4.41 UnInstallPipeline()

virtual void vtkImageColorViewer::UnInstallPipeline () [protected], [virtual]

12.393.4.42 UpdateDisplayExtent()

virtual void vtkImageColorViewer::UpdateDisplayExtent () [virtual]

12.393.4.43 UpdateOrientation()

virtual void vtkImageColorViewer::UpdateOrientation () [protected], [virtual]

12.393.4.44 vtkBooleanMacro()

vtkImageColorViewer::vtkBooleanMacro (
 OffScreenRendering ,
 int)

12.393.4.45 vtkGetMacro() [1/2]

vtkImageColorViewer::vtkGetMacro (
 [Slice](#) ,
 int)

References [Slice](#).

12.393.4.46 vtkGetMacro() [2/2]

vtkImageColorViewer::vtkGetMacro (
 [SliceOrientation](#) ,
 int)

References [SliceOrientation](#).

12.393.4.47 vtkGetObjectMacro() [1/5]

vtkImageColorViewer::vtkGetObjectMacro (
 [ImageActor](#) ,
 vtkImageActor)

References [ImageActor](#).

12.393.4.48 vtkGetObjectMacro() [2/5]

vtkImageColorViewer::vtkGetObjectMacro (
 [InteractorStyle](#) ,
 vtkInteractorStyleImage)

References [InteractorStyle](#).

12.393.4.49 vtkGetObjectMacro() [3/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    Renderer ,
    vtkRenderer )
```

References [Renderer](#).

12.393.4.50 vtkGetObjectMacro() [4/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    RenderWindow ,
    vtkRenderWindow )
```

References [RenderWindow](#).

12.393.4.51 vtkGetObjectMacro() [5/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    WindowLevel ,
    vtkImageMapToWindowLevelColors2 )
```

References [WindowLevel](#).

12.393.4.52 vtkTypeMacro()

```
vtkImageColorViewer::vtkTypeMacro (
    vtkImageColorViewer ,
    vtkObject )
```

References [vtkImageColorViewer\(\)](#).

12.393.5 Friends And Related Symbol Documentation

12.393.5.1 vtkImageColorViewerCallback

friend class vtkImageColorViewerCallback [friend]

References [vtkImageColorViewer\(\)](#), and [vtkImageColorViewerCallback](#).

Referenced by [vtkImageColorViewerCallback](#).

12.393.6 Member Data Documentation

12.393.6.1 FirstRender

int vtkImageColorViewer::FirstRender [protected]

12.393.6.2 ImageActor

`vtkImageActor* vtkImageColorViewer::ImageActor` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.393.6.3 Interactor

`vtkRenderWindowInteractor* vtkImageColorViewer::Interactor` [protected]

12.393.6.4 InteractorStyle

`vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.393.6.5 OverlayImageActor

`vtkImageActor* vtkImageColorViewer::OverlayImageActor` [protected]

12.393.6.6 Renderer

`vtkRenderer* vtkImageColorViewer::Renderer` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.393.6.7 RenderWindow

`vtkRenderWindow* vtkImageColorViewer::RenderWindow` [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

12.393.6.8 Slice

`int vtkImageColorViewer::Slice` [protected]

Referenced by [vtkGetMacro\(\)](#).

12.393.6.9 SliceOrientation

`int vtkImageColorViewer::SliceOrientation` [protected]

Referenced by [vtkGetMacro\(\)](#).

12.393.6.10 WindowLevel

[vtkImageMapToWindowLevelColors2*](#) vtkImageColorViewer::WindowLevel [protected]

Referenced by [vtkGetObjectMacro\(\)](#).

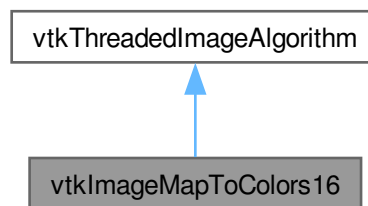
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

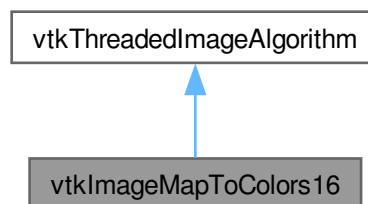
12.394 vtkImageMapToColors16 Class Reference

```
#include <vtkImageMapToColors16.h>
```

Inheritance diagram for vtkImageMapToColors16:



Collaboration diagram for vtkImageMapToColors16:



Public Member Functions

- virtual unsigned long [GetMTime](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetLookupTable](#) (vtkScalarsToColors *)
- void [SetOutputFormatToLuminance](#) ()
- void [SetOutputFormatToLuminanceAlpha](#) ()
- void [SetOutputFormatToRGB](#) ()
- void [SetOutputFormatToRGBA](#) ()
- [vtkBooleanMacro](#) (PassAlphaToOutput, int)
- [vtkGetMacro](#) (ActiveComponent, int)
- [vtkGetMacro](#) (OutputFormat, int)
- [vtkGetMacro](#) (PassAlphaToOutput, int)
- [vtkGetObjectMacro](#) (LookupTable, vtkScalarsToColors)
- [vtkSetMacro](#) (ActiveComponent, int)
- [vtkSetMacro](#) (OutputFormat, int)
- [vtkSetMacro](#) (PassAlphaToOutput, int)
- [vtkTypeMacro](#) (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static [vtkImageMapToColors16 * New](#) ()

Protected Member Functions

- [vtkImageMapToColors16](#) ()
- [~vtkImageMapToColors16](#) ()
- virtual int [RequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int [RequestInformation](#) (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void [ThreadedRequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- int [ActiveComponent](#)
- int [DataWasPassed](#)
- vtkScalarsToColors * [LookupTable](#)
- int [OutputFormat](#)
- int [PassAlphaToOutput](#)

12.394.1 Constructor & Destructor Documentation

12.394.1.1 [vtkImageMapToColors16\(\)](#)

[vtkImageMapToColors16::vtkImageMapToColors16](#) () [protected]

Referenced by [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.394.1.2 ~vtkImageMapToColors16()

vtkImageMapToColors16::~vtkImageMapToColors16 () [protected]

12.394.2 Member Function Documentation

12.394.2.1 GetMTime()

virtual unsigned long vtkImageMapToColors16::GetMTime () [virtual]

Referenced by [vtkGetMacro\(\)](#).

12.394.2.2 New()

[vtkImageMapToColors16](#) * vtkImageMapToColors16::New () [static]

References [vtkImageMapToColors16\(\)](#).

12.394.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.394.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector) [protected], [virtual]
```

12.394.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

12.394.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
    vtkScalarsToColors * ) [virtual]
```

12.394.2.7 SetOutputFormatToLuminance()

void vtkImageMapToColors16::SetOutputFormatToLuminance () [inline]

References [OutputFormat](#).

12.394.2.8 SetOutputFormatToLuminanceAlpha()

void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha () [inline]

References [OutputFormat](#).

12.394.2.9 SetOutputFormatToRGB()

void vtkImageMapToColors16::SetOutputFormatToRGB () [inline]

References [OutputFormat](#).

12.394.2.10 SetOutputFormatToRGBA()

void vtkImageMapToColors16::SetOutputFormatToRGBA () [inline]

References [OutputFormat](#).

12.394.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id) [protected]
```

12.394.2.12 vtkBooleanMacro()

```
vtkImageMapToColors16::vtkBooleanMacro (
    PassAlphaToOutput ,
    int )
```

References [PassAlphaToOutput](#).

12.394.2.13 vtkGetMacro() [1/3]

vtkImageMapToColors16::vtkGetMacro (
 [ActiveComponent](#) ,
 int)

References [ActiveComponent](#).

12.394.2.14 vtkGetMacro() [2/3]

vtkImageMapToColors16::vtkGetMacro (
 [OutputFormat](#) ,
 int)

References [OutputFormat](#).

12.394.2.15 vtkGetMacro() [3/3]

vtkImageMapToColors16::vtkGetMacro (
 [PassAlphaToOutput](#) ,
 int)

References [GetMTime\(\)](#), and [PassAlphaToOutput](#).

12.394.2.16 vtkGetObjectMacro()

vtkImageMapToColors16::vtkGetObjectMacro (
 [LookupTable](#) ,
 vtkScalarsToColors)

References [LookupTable](#).

12.394.2.17 vtkSetMacro() [1/3]

vtkImageMapToColors16::vtkSetMacro (
 [ActiveComponent](#) ,
 int)

References [ActiveComponent](#).

12.394.2.18 vtkSetMacro() [2/3]

vtkImageMapToColors16::vtkSetMacro (
 [OutputFormat](#) ,
 int)

References [OutputFormat](#).

12.394.2.19 vtkSetMacro() [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
    PassAlphaToOutput ,
    int )
```

References [PassAlphaToOutput](#).

12.394.2.20 vtkTypeMacro()

```
vtkImageMapToColors16::vtkTypeMacro (
    vtkImageMapToColors16 ,
    vtkThreadedImageAlgorithm )
```

References [vtkImageMapToColors16\(\)](#).

12.394.3 Member Data Documentation

12.394.3.1 ActiveComponent

```
int vtkImageMapToColors16::ActiveComponent [protected]
```

Referenced by [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.394.3.2 DataWasPassed

```
int vtkImageMapToColors16::DataWasPassed [protected]
```

12.394.3.3 LookupTable

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
```

Referenced by [vtkGetObjectMacro\(\)](#).

12.394.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat [protected]
```

Referenced by [SetOutputFormatToLuminance\(\)](#), [SetOutputFormatToLuminanceAlpha\(\)](#), [SetOutputFormatToRGB\(\)](#), [SetOutputFormatToRGBA\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.394.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput [protected]
```

Referenced by [vtkBooleanMacro\(\)](#), [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

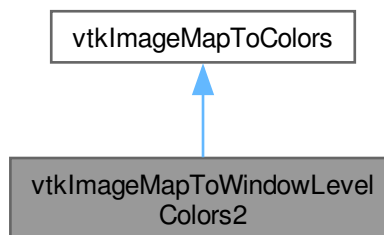
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

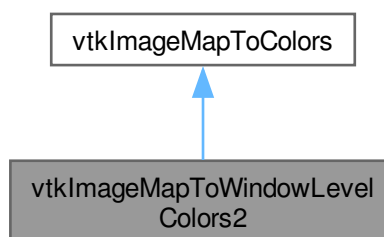
12.395 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetMacro](#) ([Level](#), double)
- [vtkGetMacro](#) ([Window](#), double)
- [vtkSetMacro](#) ([Level](#), double)
- [vtkSetMacro](#) ([Window](#), double)
- [vtkTypeMacro](#) ([vtkImageMapToWindowLevelColors2](#), vtkImageMapToColors)

Static Public Member Functions

- static [vtkImageMapToWindowLevelColors2](#) * [New](#) ()

Protected Member Functions

- [vtkImageMapToWindowLevelColors2](#) ()
- [~vtkImageMapToWindowLevelColors2](#) ()
- virtual int [RequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- virtual int [RequestInformation](#) (vtkInformation *, vtkInformationVector **, vtkInformationVector *)
- void [ThreadedRequestData](#) (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int extent[6], int id)

Protected Attributes

- double [Level](#)
- double [Window](#)

12.395.1 Constructor & Destructor Documentation

12.395.1.1 [vtkImageMapToWindowLevelColors2](#)()

[vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2](#) () [protected]

Referenced by [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.395.1.2 [~vtkImageMapToWindowLevelColors2](#)()

[vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2](#) () [protected]

12.395.2 Member Function Documentation

12.395.2.1 New()

[vtkImageMapToWindowLevelColors2](#) * [vtkImageMapToWindowLevelColors2::New](#) () [static]

References [vtkImageMapToWindowLevelColors2\(\)](#).

12.395.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.395.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector) [protected], [virtual]
```

12.395.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

12.395.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id) [protected]
```

12.395.2.6 vtkGetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Level ,
    double )
```

References [Level](#).

12.395.2.7 `vtkGetMacro()` [2/2]

`vtkImageMapToWindowLevelColors2::vtkGetMacro (`
 [Window](#) ,
 double)

References [Window](#).

12.395.2.8 `vtkSetMacro()` [1/2]

`vtkImageMapToWindowLevelColors2::vtkSetMacro (`
 [Level](#) ,
 double)

References [Level](#).

12.395.2.9 `vtkSetMacro()` [2/2]

`vtkImageMapToWindowLevelColors2::vtkSetMacro (`
 [Window](#) ,
 double)

References [Window](#).

12.395.2.10 `vtkTypeMacro()`

`vtkImageMapToWindowLevelColors2::vtkTypeMacro (`
 [vtkImageMapToWindowLevelColors2](#) ,
 [vtkImageMapToColors](#))

References [vtkImageMapToWindowLevelColors2\(\)](#).

12.395.3 Member Data Documentation

12.395.3.1 `Level`

double `vtkImageMapToWindowLevelColors2::Level` [protected]

Referenced by [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

12.395.3.2 `Window`

double `vtkImageMapToWindowLevelColors2::Window` [protected]

Referenced by [vtkGetMacro\(\)](#), and [vtkSetMacro\(\)](#).

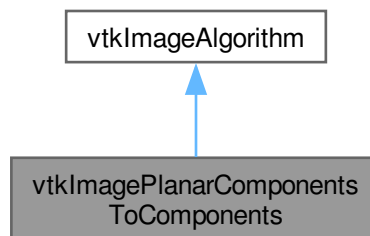
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

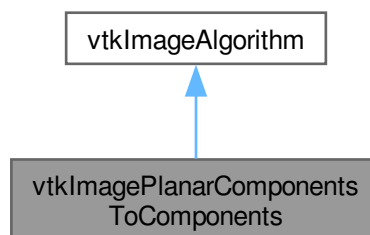
12.396 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void `PrintSelf` (ostream &os, vtkIndent indent)
- `vtkTypeMacro` (`vtkImagePlanarComponentsToComponents`, vtkImageAlgorithm)

Static Public Member Functions

- static `vtkImagePlanarComponentsToComponents * New` ()

Protected Member Functions

- [vtkImagePlanarComponentsToComponents](#) ()
- [~vtkImagePlanarComponentsToComponents](#) ()
- virtual int [RequestData](#) (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

12.396.1 Constructor & Destructor Documentation

12.396.1.1 [vtkImagePlanarComponentsToComponents](#)()

[vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents](#) () [protected]

Referenced by [New\(\)](#), [RequestData\(\)](#), and [vtkTypeMacro\(\)](#).

12.396.1.2 [~vtkImagePlanarComponentsToComponents](#)()

[vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents](#) () [inline], [protected]

12.396.2 Member Function Documentation

12.396.2.1 [New\(\)](#)

[vtkImagePlanarComponentsToComponents](#) * [vtkImagePlanarComponentsToComponents::New](#) () [static]

References [vtkImagePlanarComponentsToComponents\(\)](#).

12.396.2.2 [PrintSelf\(\)](#)

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.396.2.3 [RequestData\(\)](#)

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

References [vtkImagePlanarComponentsToComponents\(\)](#).

12.396.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
    vtkImagePlanarComponentsToComponents ,
    vtkImageAlgorithm )
```

References [vtkImagePlanarComponentsToComponents\(\)](#).

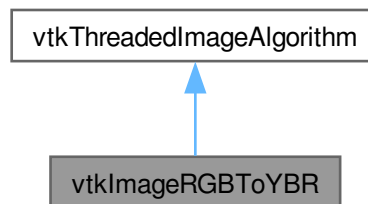
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

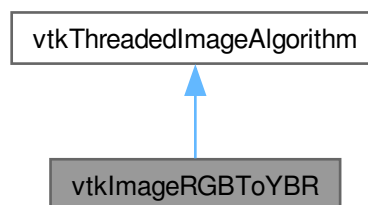
12.397 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkImageRGBToYBR](#), vtkThreadedImageAlgorithm)

Static Public Member Functions

- static [vtkImageRGBToYBR](#) * [New](#) ()

Protected Member Functions

- [vtkImageRGBToYBR](#) ()
- [~vtkImageRGBToYBR](#) ()
- void [ThreadedExecute](#) (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

12.397.1 Constructor & Destructor Documentation

12.397.1.1 [vtkImageRGBToYBR\(\)](#)

[vtkImageRGBToYBR::vtkImageRGBToYBR](#) () [protected]

Referenced by [New\(\)](#), [ThreadedExecute\(\)](#), and [vtkTypeMacro\(\)](#).

12.397.1.2 [~vtkImageRGBToYBR\(\)](#)

[vtkImageRGBToYBR::~~vtkImageRGBToYBR](#) () [inline], [protected]

12.397.2 Member Function Documentation

12.397.2.1 [New\(\)](#)

[vtkImageRGBToYBR](#) * [vtkImageRGBToYBR::New](#) () [static]

References [vtkImageRGBToYBR\(\)](#).

12.397.2.2 [PrintSelf\(\)](#)

```
void vtkImageRGBToYBR::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.397.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id) [protected]
```

References [vtkImageRGBToYBR\(\)](#).

12.397.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
    vtkImageRGBToYBR ,
    vtkThreadedImageAlgorithm )
```

References [vtkImageRGBToYBR\(\)](#).

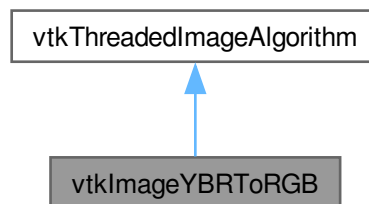
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

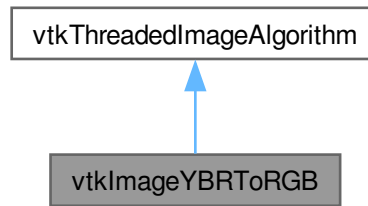
12.398 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeMacro](#) ([vtkImageYBRToRGB](#), vtkThreadedImageAlgorithm)

Static Public Member Functions

- static [vtkImageYBRToRGB * New](#) ()

Protected Member Functions

- [vtkImageYBRToRGB](#) ()
- [~vtkImageYBRToRGB](#) ()
- void [ThreadedExecute](#) (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

12.398.1 Constructor & Destructor Documentation

12.398.1.1 [vtkImageYBRToRGB](#)()

`vtkImageYBRToRGB::vtkImageYBRToRGB ()` [protected]

Referenced by [New\(\)](#), [ThreadedExecute\(\)](#), and [vtkTypeMacro\(\)](#).

12.398.1.2 [~vtkImageYBRToRGB](#)()

`vtkImageYBRToRGB::~~vtkImageYBRToRGB ()` [inline], [protected]

12.398.2 Member Function Documentation

12.398.2.1 New()

[vtkImageYBRToRGB](#) * [vtkImageYBRToRGB::New](#) () [static]

References [vtkImageYBRToRGB\(\)](#).

12.398.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.398.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id) [protected]
```

References [vtkImageYBRToRGB\(\)](#).

12.398.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
    vtkImageYBRToRGB ,
    vtkThreadedImageAlgorithm )
```

References [vtkImageYBRToRGB\(\)](#).

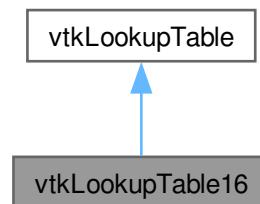
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

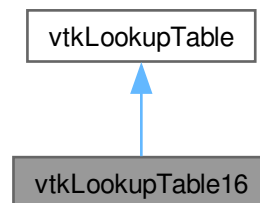
12.399 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

Inheritance diagram for vtkLookupTable16:



Collaboration diagram for vtkLookupTable16:



Public Member Functions

- void [Build](#) ()
- unsigned short * [GetPointer](#) (const vtkIdType id)
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- void [SetNumberOfTableValues](#) (vtkIdType number)
- [vtkTypeMacro](#) ([vtkLookupTable16](#), vtkLookupTable)
- unsigned char * [WritePointer](#) (const vtkIdType id, const int number)

Static Public Member Functions

- static [vtkLookupTable16](#) * [New](#) ()

Protected Member Functions

- [vtkLookupTable16](#) (int size=256, int ext=256)
- [~vtkLookupTable16](#) ()
- void [MapScalarsThroughTable2](#) (void *input, unsigned char *output, int inputDataType, int number↵OfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray * [Table16](#)

12.399.1 Constructor & Destructor Documentation

12.399.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
    int size = 256,
    int ext = 256) [protected]
```

Referenced by [MapScalarsThroughTable2\(\)](#), [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.399.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~~vtkLookupTable16 () [protected]
```

12.399.2 Member Function Documentation

12.399.2.1 Build()

```
void vtkLookupTable16::Build ()
```

12.399.2.2 GetPointer()

```
unsigned short * vtkLookupTable16::GetPointer (
    const vtkIdType id) [inline]
```

References [Table16](#).

12.399.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (  
    void * input,  
    unsigned char * output,  
    int inputDataType,  
    int numberOfValues,  
    int inputIncrement,  
    int outputFormat) [protected]
```

References [vtkLookupTable16\(\)](#).

12.399.2.4 New()

```
vtkLookupTable16 * vtkLookupTable16::New () [static]
```

References [vtkLookupTable16\(\)](#).

12.399.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (  
    ostream & os,  
    vtkIndent indent)
```

12.399.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (  
    vtkIdType number)
```

References [WritePointer\(\)](#).

12.399.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (  
    vtkLookupTable16 ,  
    vtkLookupTable )
```

References [vtkLookupTable16\(\)](#).

12.399.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (  
    const vtkIdType id,  
    const int number) [inline]
```

References [Table16](#).

Referenced by [SetNumberOfTableValues\(\)](#).

12.399.3 Member Data Documentation

12.399.3.1 Table16

vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by [GetPointer\(\)](#), and [WritePointer\(\)](#).

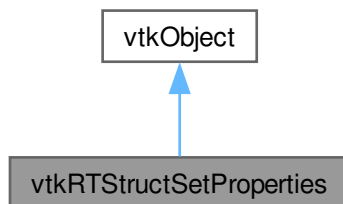
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

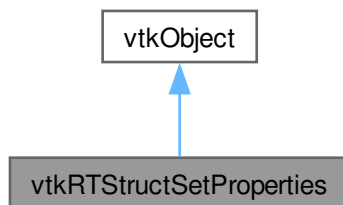
12.400 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Inheritance diagram for vtkRTStructSetProperties:



Collaboration diagram for vtkRTStructSetProperties:



Public Member Functions

- void [AddContourReferencedFrameOfReference](#) (vtkIdType pdnum, const char *classuid, const char *instanceuid)
- void [AddReferencedFrameOfReference](#) (const char *classuid, const char *instanceuid)
- void [AddStructureSetROI](#) (int roinumber, const char *reframerefuid, const char *roiname, const char *ROIGenerationAlgorithm, const char *ROIDescription=0)
- void [AddStructureSetROIObservation](#) (int refnumber, int observationnumber, const char *rtroiinterpretedtype, const char *roiinterpreter, const char *roiobservationlabel=0)
- virtual void [Clear](#) ()
- virtual void [DeepCopy](#) (vtkRTStructSetProperties *p)
- const char * [GetContourReferencedFrameOfReferenceClassUID](#) (vtkIdType pdnum, vtkIdType id)
- const char * [GetContourReferencedFrameOfReferenceInstanceUID](#) (vtkIdType pdnum, vtkIdType id)
- vtkIdType [GetNumberOfContourReferencedFrameOfReferences](#) ()
- vtkIdType [GetNumberOfContourReferencedFrameOfReferences](#) (vtkIdType pdnum)
- vtkIdType [GetNumberOfReferencedFrameOfReferences](#) ()
- vtkIdType [GetNumberOfStructureSetROIs](#) ()
- const char * [GetReferencedFrameOfReferenceClassUID](#) (vtkIdType id)
- const char * [GetReferencedFrameOfReferenceInstanceUID](#) (vtkIdType id)
- int [GetStructureSetObservationNumber](#) (vtkIdType id)
- const char * [GetStructureSetROIDescription](#) (vtkIdType id)
- const char * [GetStructureSetROIGenerationAlgorithm](#) (vtkIdType)
- const char * [GetStructureSetROIName](#) (vtkIdType)
- int [GetStructureSetROINumber](#) (vtkIdType id)
- const char * [GetStructureSetROIObservationLabel](#) (vtkIdType id)
- const char * [GetStructureSetROIRefFrameRefUID](#) (vtkIdType)
- const char * [GetStructureSetRTROIInterpretedType](#) (vtkIdType id)
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetStringMacro](#) (ReferenceFrameOfReferenceUID)
- [vtkGetStringMacro](#) (ReferenceSeriesInstanceUID)
- [vtkGetStringMacro](#) (SeriesInstanceUID)
- [vtkGetStringMacro](#) (SOPInstanceUID)
- [vtkGetStringMacro](#) (StructureSetDate)
- [vtkGetStringMacro](#) (StructureSetLabel)
- [vtkGetStringMacro](#) (StructureSetName)
- [vtkGetStringMacro](#) (StructureSetTime)
- [vtkGetStringMacro](#) (StudyInstanceUID)
- [vtkSetStringMacro](#) (ReferenceFrameOfReferenceUID)
- [vtkSetStringMacro](#) (ReferenceSeriesInstanceUID)
- [vtkSetStringMacro](#) (SeriesInstanceUID)
- [vtkSetStringMacro](#) (SOPInstanceUID)
- [vtkSetStringMacro](#) (StructureSetDate)
- [vtkSetStringMacro](#) (StructureSetLabel)
- [vtkSetStringMacro](#) (StructureSetName)
- [vtkSetStringMacro](#) (StructureSetTime)
- [vtkSetStringMacro](#) (StudyInstanceUID)
- [vtkTypeMacro](#) (vtkRTStructSetProperties, vtkObject)

Static Public Member Functions

- static [vtkRTStructSetProperties](#) * [New](#) ()

Protected Member Functions

- [vtkRTStructSetProperties \(\)](#)
- [~vtkRTStructSetProperties \(\)](#)

Protected Attributes

- vtkRTStructSetPropertiesInternals * [Internals](#)
- char * [ReferenceFrameOfReferenceUID](#)
- char * [ReferenceSeriesInstanceUID](#)
- char * [SeriesInstanceUID](#)
- char * [SOPInstanceUID](#)
- char * [StructureSetDate](#)
- char * [StructureSetLabel](#)
- char * [StructureSetName](#)
- char * [StructureSetTime](#)
- char * [StudyInstanceUID](#)

12.400.1 Detailed Description

Examples

[GenerateRTSTRUCT.cxx](#).

12.400.2 Constructor & Destructor Documentation

12.400.2.1 vtkRTStructSetProperties()

`vtkRTStructSetProperties::vtkRTStructSetProperties ()` [protected]

Referenced by [DeepCopy\(\)](#), [New\(\)](#), and [vtkTypeMacro\(\)](#).

12.400.2.2 ~vtkRTStructSetProperties()

`vtkRTStructSetProperties::~~vtkRTStructSetProperties ()` [protected]

12.400.3 Member Function Documentation

12.400.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
    vtkIdType pdnum,
    const char * classuid,
    const char * instanceuid)
```

12.400.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
    const char * classuid,
    const char * instanceuid)
```

12.400.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
    int roinumber,
    const char * refframerefid,
    const char * roiname,
    const char * ROIGenerationAlgorithm,
    const char * ROIDescription = 0)
```

12.400.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
    int refnumber,
    int observationnumber,
    const char * rtroiinterpretedtype,
    const char * roiinterpreter,
    const char * roiobservationlabel = 0)
```

12.400.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear () [virtual]
```

12.400.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
    vtkRTStructSetProperties * p) [virtual]
```

References [vtkRTStructSetProperties\(\)](#).

12.400.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
    vtkIdType pdnum,
    vtkIdType id)
```

12.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
    vtkIdType pdnum,
    vtkIdType id)
```


12.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ()

12.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
 vtkIdType pdnum)

12.400.3.11 GetNumberOfReferencedFrameOfReferences()

vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ()

12.400.3.12 GetNumberOfStructureSetROIs()

vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ()

12.400.3.13 GetReferencedFrameOfReferenceClassUID()

const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
 vtkIdType id)

12.400.3.14 GetReferencedFrameOfReferenceInstanceUID()

const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
 vtkIdType id)

12.400.3.15 GetStructureSetObservationNumber()

int vtkRTStructSetProperties::GetStructureSetObservationNumber (
 vtkIdType id)

12.400.3.16 GetStructureSetROIDescription()

const char * vtkRTStructSetProperties::GetStructureSetROIDescription (
 vtkIdType id)

12.400.3.17 GetStructureSetROIGenerationAlgorithm()

const char * vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
 vtkIdType)

12.400.3.18 GetStructureSetROIName()

```
const char * vtkRTStructSetProperties::GetStructureSetROIName (
    vtkIdType )
```

12.400.3.19 GetStructureSetROINumber()

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
    vtkIdType id)
```

12.400.3.20 GetStructureSetROIObservationLabel()

```
const char * vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
    vtkIdType id)
```

12.400.3.21 GetStructureSetROIRefFrameRefUID()

```
const char * vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
    vtkIdType )
```

12.400.3.22 GetStructureSetRTROIInterpretedType()

```
const char * vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
    vtkIdType id)
```

12.400.3.23 New()

```
vtkRTStructSetProperties * vtkRTStructSetProperties::New () [static]
```

Examples

[GenerateRTSTRUCT.cxx](#).

References [vtkRTStructSetProperties\(\)](#).

12.400.3.24 PrintSelf()

```
void vtkRTStructSetProperties::PrintSelf (
    ostream & os,
    vtkIndent indent)
```

12.400.3.25 vtkGetStringMacro() [1/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [ReferenceFrameOfReferenceUID](#))

References [ReferenceFrameOfReferenceUID](#).

12.400.3.26 vtkGetStringMacro() [2/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [ReferenceSeriesInstanceUID](#))

References [ReferenceSeriesInstanceUID](#).

12.400.3.27 vtkGetStringMacro() [3/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [SeriesInstanceUID](#))

References [SeriesInstanceUID](#).

12.400.3.28 vtkGetStringMacro() [4/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [SOPInstanceUID](#))

References [SOPInstanceUID](#).

12.400.3.29 vtkGetStringMacro() [5/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [StructureSetDate](#))

References [StructureSetDate](#).

12.400.3.30 vtkGetStringMacro() [6/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [StructureSetLabel](#))

References [StructureSetLabel](#).

12.400.3.31 vtkGetStringMacro() [7/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [StructureSetName](#))

References [StructureSetName](#).

12.400.3.32 vtkGetStringMacro() [8/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [StructureSetTime](#))

References [StructureSetTime](#).

12.400.3.33 vtkGetStringMacro() [9/9]

vtkRTStructSetProperties::vtkGetStringMacro (
 [StudyInstanceUID](#))

References [StudyInstanceUID](#).

12.400.3.34 vtkSetStringMacro() [1/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [ReferenceFrameOfReferenceUID](#))

References [ReferenceFrameOfReferenceUID](#).

12.400.3.35 vtkSetStringMacro() [2/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [ReferenceSeriesInstanceUID](#))

References [ReferenceSeriesInstanceUID](#).

12.400.3.36 vtkSetStringMacro() [3/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [SeriesInstanceUID](#))

References [SeriesInstanceUID](#).

12.400.3.37 vtkSetStringMacro() [4/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [SOPInstanceUID](#))

References [SOPInstanceUID](#).

12.400.3.38 vtkSetStringMacro() [5/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [StructureSetDate](#))

References [StructureSetDate](#).

12.400.3.39 vtkSetStringMacro() [6/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [StructureSetLabel](#))

References [StructureSetLabel](#).

12.400.3.40 vtkSetStringMacro() [7/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [StructureSetName](#))

References [StructureSetName](#).

12.400.3.41 vtkSetStringMacro() [8/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [StructureSetTime](#))

References [StructureSetTime](#).

12.400.3.42 vtkSetStringMacro() [9/9]

vtkRTStructSetProperties::vtkSetStringMacro (
 [StudyInstanceUID](#))

References [StudyInstanceUID](#).

12.400.3.43 vtkTypeMacro()

```
vtkRTStructSetProperties::vtkTypeMacro (
    vtkRTStructSetProperties ,
    vtkObject )
```

References [vtkRTStructSetProperties\(\)](#).

12.400.4 Member Data Documentation

12.400.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]
```

12.400.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]
```

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]
```

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID [protected]
```

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID [protected]
```

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate [protected]
```

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.7 StructureSetLabel

char* vtkRTStructSetProperties::StructureSetLabel [protected]

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.8 StructureSetName

char* vtkRTStructSetProperties::StructureSetName [protected]

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.9 StructureSetTime

char* vtkRTStructSetProperties::StructureSetTime [protected]

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

12.400.4.10 StudyInstanceUID

char* vtkRTStructSetProperties::StudyInstanceUID [protected]

Referenced by [vtkGetStringMacro\(\)](#), and [vtkSetStringMacro\(\)](#).

The documentation for this class was generated from the following file:

- [vtkRTStructSetProperties.h](#)

12.401 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- [Waveform](#) ()=default

12.401.1 Detailed Description

[Waveform](#) class.

12.401.2 Constructor & Destructor Documentation

12.401.2.1 Waveform()

gdcm::Waveform::Waveform () [default]

The documentation for this class was generated from the following file:

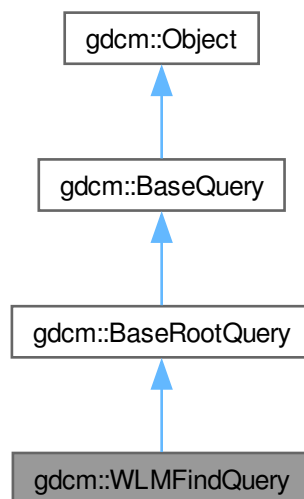
- [gdcmWaveform.h](#)

12.402 gdcm::WLMFindQuery Class Reference

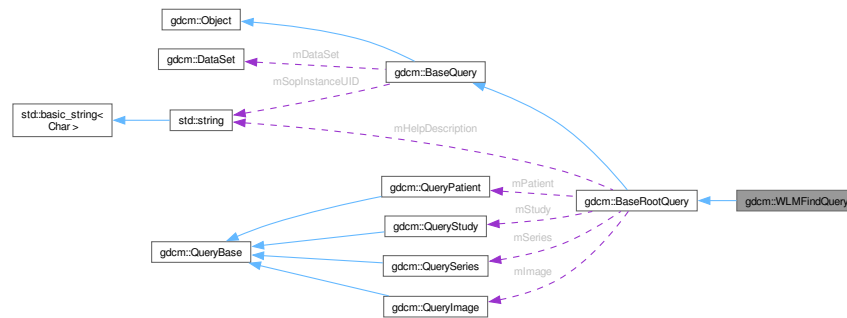
PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



Public Member Functions

- [WLMFindQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const override
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel) override
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel) override
- bool [ValidateQuery](#) (bool inStrict=true) const override

Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- [~BaseRootQuery](#) () override=default
- [EQueryLevel GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)

Public Member Functions inherited from [gdcm::BaseQuery](#)

- [~BaseQuery](#) () override
- void [AddQueryDataSet](#) (const [DataSet](#) &ds)
- [DataSet & GetQueryDataSet](#) ()
- [DataSet](#) const & [GetQueryDataSet](#) () const
- Set/Get the internal representation of the query as a [DataSet](#).
- [std::string GetSOPInstanceUID](#) () const
- void [Print](#) (std::ostream &os) const override
- void [SetSearchParameter](#) (const std::string &inKeyword, const std::string &inValue)
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const std::string &inValue)
- void [SetSOPInstanceUID](#) (const std::string &iSopInstanceUID)
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

Public Member Functions inherited from [gdcm::Object](#)

- [Object](#) ()
- [Object](#) (const [Object](#) &)
- Special requirement for copy/cstor, assignment operator.
- virtual [~Object](#) ()
- void [operator=](#) (const [Object](#) &)

Protected Member Functions

- [DataSet](#) [GetValidDataSet](#) () const

Protected Member Functions inherited from [gdcm::BaseRootQuery](#)

- [BaseRootQuery](#) ()

Protected Member Functions inherited from [gdcm::BaseQuery](#)

- [BaseQuery](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &in↵ Value)
- bool [ValidDataSet](#) (const [DataSet](#) &dataSetToValid, const [DataSet](#) &dataSetReference) const

Protected Member Functions inherited from [gdcm::Object](#)

- void [Register](#) ()
- void [UnRegister](#) ()

Friends

- class [QueryFactory](#)

Additional Inherited Members

Static Public Member Functions inherited from [gdcm::BaseRootQuery](#)

- static [QueryBase](#) * [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char *str)
- static const char * [GetQueryLevelString](#) ([EQueryLevel](#) ql)

Protected Attributes inherited from [gdcm::BaseRootQuery](#)

- std::string [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

Protected Attributes inherited from [gdcm::BaseQuery](#)

- [DataSet](#) [mDataSet](#)
- std::string [mSopInstanceUID](#)

12.402.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

12.402.2 Constructor & Destructor Documentation

12.402.2.1 WLMFindQuery()

gdcm::WLMFindQuery::WLMFindQuery ()

12.402.3 Member Function Documentation

12.402.3.1 GetAbstractSyntaxUID()

[UIDs::TSName](#) gdcm::WLMFindQuery::GetAbstractSyntaxUID () const [override], [virtual]

Implements [gdcm::BaseQuery](#).

12.402.3.2 GetTagListByLevel()

std::vector< [Tag](#) > gdcm::WLMFindQuery::GetTagListByLevel (
const [EQueryLevel](#) & inQueryLevel) [override], [virtual]

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements [gdcm::BaseRootQuery](#).

12.402.3.3 GetValidDataSet()

[DataSet](#) gdcm::WLMFindQuery::GetValidDataSet () const [protected]

12.402.3.4 InitializeDataSet()

void gdcm::WLMFindQuery::InitializeDataSet (
const [EQueryLevel](#) & inQueryLevel) [override], [virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements [gdcm::BaseRootQuery](#).

12.402.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
    bool inStrict = true) const    [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements [gdcm::BaseRootQuery](#).

12.402.4 Friends And Related Symbol Documentation

12.402.4.1 QueryFactory

```
friend class QueryFactory    [friend]
```

References [QueryFactory](#).

Referenced by [QueryFactory](#).

The documentation for this class was generated from the following file:

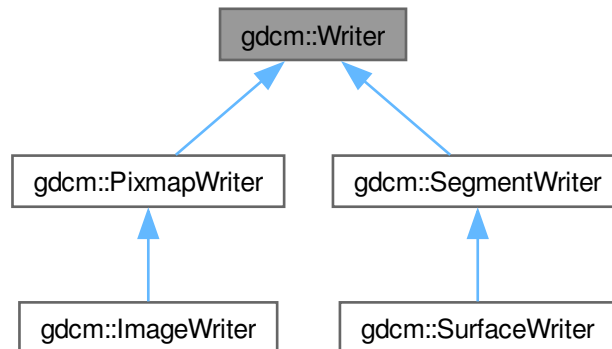
- [gdcmWLMFindQuery.h](#)

12.403 gdcm::Writer Class Reference

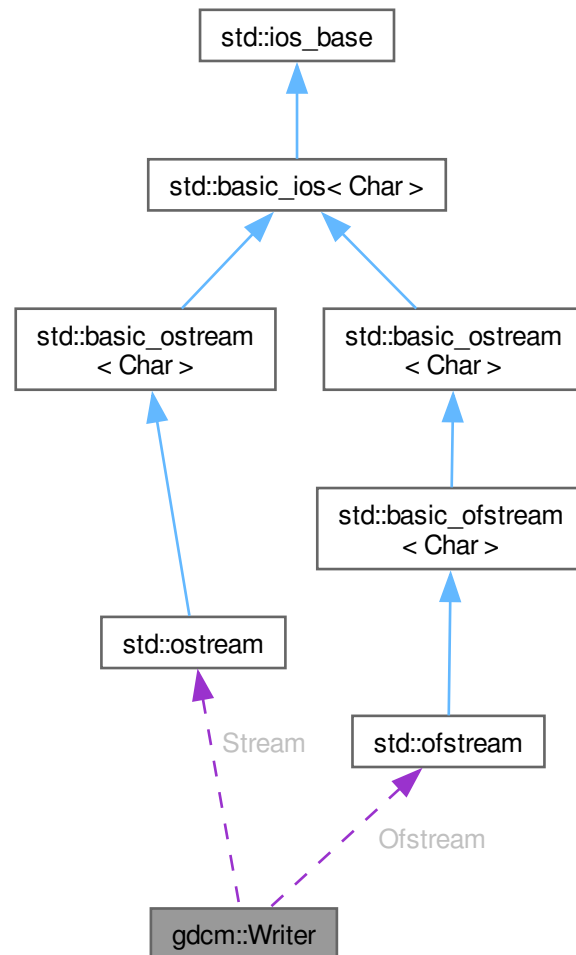
[Writer](#) ala DOM (Document [Object](#) Model).

```
#include <gdcmWriter.h>
```

Inheritance diagram for gdcm::Writer:



Collaboration diagram for gdcm::Writer:



Public Member Functions

- [Writer](#) ()
- virtual [~Writer](#) ()
- void [CheckFileMetaInformationOff](#) ()
- void [CheckFileMetaInformationOn](#) ()
- [File](#) & [GetFile](#) ()
- void [SetCheckFileMetaInformation](#) (bool b)
Undocumented function, do not use (= leave default).
- void [SetFile](#) (const [File](#) &f)
Set/Get the DICOM file ([DataSet](#) + Header).

- void [SetFileName](#) (const char *filename__native)
Set the filename of DICOM file to write:
- void [SetStream](#) (std::ostream &output__stream)
Set user ostream buffer.
- virtual bool [Write](#) ()
Main function to tell the writer to write.

Protected Member Functions

- bool [GetCheckFileMetaInformation](#) () const
- std::ostream * [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

Protected Attributes

- std::ofstream * [Ofstream](#)
- std::ostream * [Stream](#)

Friends

- class [StreamImageWriter](#)

12.403.1 Detailed Description

[Writer](#) ala DOM (Document [Object](#) Model).

This class is a non-validating writer, it will only performs well- formedness check only.

Detailed description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for [Item](#) start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see [gdcm::FileMetaInformation](#))
- Zero value for [Item](#) Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (guaranteed by design of internals)
- 32bits [VR](#) will be rewritten with 00

Warning

[gdcm::Writer](#) cannot write a [DataSet](#) if no SOP Instance UID (0008,0018) is found, unless a [DICOMDIR](#) is being written out

See also

[Reader DataSet File](#)

Examples

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [StreamImageReaderTest.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.403.2 Constructor & Destructor Documentation

12.403.2.1 Writer()

gdcm::Writer::Writer ()

12.403.2.2 ~Writer()

virtual gdcm::Writer::~Writer () [virtual]

12.403.3 Member Function Documentation

12.403.3.1 CheckFileMetaInformationOff()

void gdcm::Writer::CheckFileMetaInformationOff () [inline]

Examples

[CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

12.403.3.2 CheckFileMetaInformationOn()

void gdcm::Writer::CheckFileMetaInformationOn () [inline]

12.403.3.3 GetCheckFileMetaInformation()

bool gdcm::Writer::GetCheckFileMetaInformation () const [inline], [protected]

12.403.3.4 GetFile()

[File](#) & `gdcmm::Writer::GetFile ()` [inline]

Examples

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [MpegVideoInfo.cs](#), [QIDO-RS.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.403.3.5 GetStreamPtr()

`std::ostream * gdcmm::Writer::GetStreamPtr () const` [inline], [protected]

References [Stream](#).

12.403.3.6 SetCheckFileMetaInformation()

`void gdcmm::Writer::SetCheckFileMetaInformation (`
 `bool b) [inline]`

Undocumented function, do not use (= leave default).

Examples

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

12.403.3.7 SetFile()

`void gdcmm::Writer::SetFile (`
 `const File & f) [inline]`

Set/Get the DICOM file ([DataSet](#) + Header).

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateFakeRTDOSE.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.403.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
    const char * filename_native)
```

Set the filename of DICOM file to write:

Examples

[BasicAnonymizer.cs](#), [BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CompressImage.cxx](#), [CompressLossyJPEG.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DecompressImage.cs](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [MergeTwoFiles.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), [TemplateEmptyImage.cxx](#), [csa2img.cxx](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.403.3.9 SetStream()

```
void gdcm::Writer::SetStream (
    std::ostream & output_stream) [inline]
```

Set user ostream buffer.

References [Stream](#).

12.403.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
    bool b) [inline], [protected]
```

12.403.3.11 Write()

```
virtual bool gdcm::Writer::Write () [virtual]
```

Main function to tell the writer to write.

Reimplemented in [gdcm::ImageWriter](#), [gdcm::PixmapWriter](#), [gdcm::SegmentWriter](#), and [gdcm::SurfaceWriter](#).

Examples

[BasicAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [Cleaner.cs](#), [ClinicalTrialAnnotate.cxx](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GenerateDICOMDIR.cs](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [ManipulateFile.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReformatFile.cs](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

12.403.4 Friends And Related Symbol Documentation

12.403.4.1 StreamImageWriter

friend class StreamImageWriter [friend]

References [StreamImageWriter](#).

Referenced by [StreamImageWriter](#).

12.403.5 Member Data Documentation

12.403.5.1 Ofstream

std::ofstream* gdcm::Writer::Ofstream [protected]

12.403.5.2 Stream

std::ostream* gdcm::Writer::Stream [protected]

Referenced by [GetStreamPtr\(\)](#), and [SetStream\(\)](#).

The documentation for this class was generated from the following file:

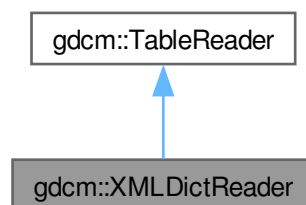
- [gdcmWriter.h](#)

12.404 gdcm::XMLDictReader Class Reference

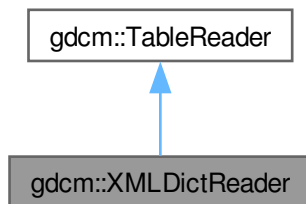
Class for representing a [XMLDictReader](#).

```
#include <gdcmXMLDictReader.h>
```

Inheritance diagram for gdcm::XMLDictReader:



Collaboration diagram for gdcm::XMLDictReader:



Public Member Functions

- [XMLDictReader](#) ()
- [~XMLDictReader](#) ()
- void [CharacterDataHandler](#) (const char *data, int length)
- void [EndElement](#) (const char *name)
- const [Dict](#) & [GetDict](#) ()
- void [StartElement](#) (const char *name, const char **atts)

Public Member Functions inherited from [gdcm::TableReader](#)

- [TableReader](#) (Defs &defs)
- virtual [~TableReader](#) ()=default
- const [Defs](#) & [GetDefs](#) () const
- const char * [GetFilename](#) ()
- void [HandleIOD](#) (const char **atts)
- void [HandleIODEntry](#) (const char **atts)
- void [HandleMacro](#) (const char **atts)
- void [HandleMacroEntry](#) (const char **atts)
- void [HandleMacroEntryDescription](#) (const char **atts)
- void [HandleModule](#) (const char **atts)
- void [HandleModuleEntry](#) (const char **atts)
- void [HandleModuleEntryDescription](#) (const char **atts)
- void [HandleModuleInclude](#) (const char **atts)
- int [Read](#) ()
- void [SetFilename](#) (const char *filename)

Protected Member Functions

- void [HandleDescription](#) (const char **atts)
- void [HandleEntry](#) (const char **atts)

12.404.1 Detailed Description

Class for representing a [XMLDictReader](#).

Note

bla Will read the DICOMV3.xml file

12.404.2 Constructor & Destructor Documentation

12.404.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ()
```

12.404.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~~XMLDictReader () [inline]
```

12.404.3 Member Function Documentation

12.404.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (  
    const char * data,  
    int length) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

12.404.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (  
    const char * name) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

12.404.3.3 GetDict()

```
const Dict & gdcm::XMLDictReader::GetDict () [inline]
```

12.404.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (  
    const char ** atts) [protected]
```

12.404.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
    const char ** atts) [protected]
```

12.404.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
    const char * name,
    const char ** atts) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

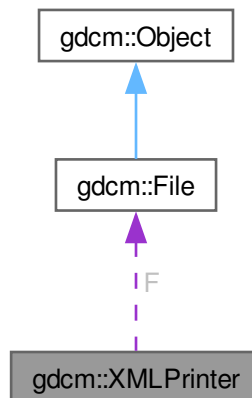
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

12.405 gdcm::XMLPrinter Class Reference

```
#include <gdcmXMLPrinter.h>
```

Collaboration diagram for gdcm::XMLPrinter:



Public Types

- enum [PrintStyles](#) {
 [OnlyUUID](#) = 0 ,
 [LOADBULKDATA](#) = 1 }

Public Member Functions

- [XMLPrinter](#) ()
- virtual [~XMLPrinter](#) ()
- [PrintStyles](#) [GetPrintStyle](#) () const
- virtual void [HandleBulkData](#) (const char *uuid, const [TransferSyntax](#) &ts, const char *bulkdata, size_t bulklen)
- void [Print](#) (std::ostream &os)
- void [PrintDataSet](#) (const [DataSet](#) &ds, const [TransferSyntax](#) &ts, std::ostream &os)
- void [SetFile](#) ([File](#) const &f)
- void [SetStyle](#) ([PrintStyles](#) ps)

Protected Member Functions

- [VR PrintDataElement](#) (std::ostream &os, const [Dicts](#) &dicts, const [DataSet](#) &ds, const [DataElement](#) &de, const [TransferSyntax](#) &ts)
- void [PrintSQ](#) (const [SequenceOfItems](#) *sqi, const [TransferSyntax](#) &ts, std::ostream &os)

Protected Attributes

- const [File](#) * F
- [PrintStyles](#) [PrintStyle](#)

12.405.1 Member Enumeration Documentation

12.405.1.1 PrintStyles

enum [gdcm::XMLPrinter::PrintStyles](#)

Enumerator

OnlyUUID	
LOADBULKDATA	

12.405.2 Constructor & Destructor Documentation

12.405.2.1 XMLPrinter()

[gdcm::XMLPrinter::XMLPrinter](#) ()

12.405.2.2 ~XMLPrinter()

virtual [gdcm::XMLPrinter::~XMLPrinter](#) () [virtual]

12.405.3 Member Function Documentation

12.405.3.1 GetPrintStyle()

[PrintStyles](#) gdcm::XMLPrinter::GetPrintStyle () const [inline]

References [PrintStyle](#).

12.405.3.2 HandleBulkData()

```
virtual void gdcm::XMLPrinter::HandleBulkData (  
    const char * uuid,  
    const TransferSyntax & ts,  
    const char * bulkdata,  
    size_t bulklen) [virtual]
```

Virtual function mechanism to allow application programmer to override the default mechanism for BulkData handling. By default GDCM will simply discard the BulkData and only write the UUID

12.405.3.3 Print()

```
void gdcm::XMLPrinter::Print (  
    std::ostream & os)
```

12.405.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (  
    std::ostream & os,  
    const Dicts & dicts,  
    const DataSet & ds,  
    const DataElement & de,  
    const TransferSyntax & ts) [protected]
```

12.405.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (  
    const DataSet & ds,  
    const TransferSyntax & ts,  
    std::ostream & os)
```

12.405.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (  
    const SequenceOfItems * sqi,  
    const TransferSyntax & ts,  
    std::ostream & os) [protected]
```

12.405.3.7 SetFile()

```
void gdcm::XMLPrinter::SetFile (  
    File const & f) [inline]
```

References [F](#).

12.405.3.8 SetStyle()

```
void gdcm::XMLPrinter::SetStyle (  
    PrintStyles ps) [inline]
```

References [PrintStyle](#).

12.405.4 Member Data Documentation

12.405.4.1 F

```
const File* gdcm::XMLPrinter::F [protected]
```

Referenced by [SetFile\(\)](#).

12.405.4.2 PrintStyle

```
PrintStyles gdcm::XMLPrinter::PrintStyle [protected]
```

Referenced by [GetPrintStyle\(\)](#), and [SetStyle\(\)](#).

The documentation for this class was generated from the following file:

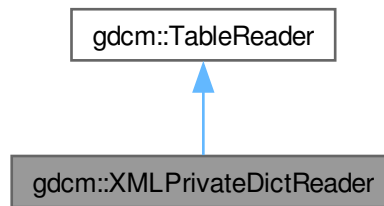
- [gdcmXMLPrinter.h](#)

12.406 gdcm::XMLPrivateDictReader Class Reference

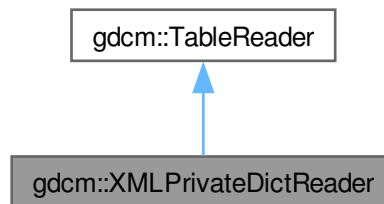
Class for representing a [XMLPrivateDictReader](#).

```
#include <gdcmXMLPrivateDictReader.h>
```

Inheritance diagram for gdcm::XMLPrivateDictReader:



Collaboration diagram for gdcm::XMLPrivateDictReader:



Public Member Functions

- [XMLPrivateDictReader](#) ()
- [~XMLPrivateDictReader](#) ()
- void [CharacterDataHandler](#) (const char *data, int length)
- void [EndElement](#) (const char *name)
- const [PrivateDict](#) & [GetPrivateDict](#) ()
- void [StartElement](#) (const char *name, const char **atts)

Public Member Functions inherited from [gdcm::TableReader](#)

- [TableReader](#) ([Defs](#) &defs)
- virtual [~TableReader](#) ()=default
- const [Defs](#) & [GetDefs](#) () const
- const char * [GetFilename](#) ()
- void [HandleIOD](#) (const char **atts)
- void [HandleIODEntry](#) (const char **atts)
- void [HandleMacro](#) (const char **atts)
- void [HandleMacroEntry](#) (const char **atts)
- void [HandleMacroEntryDescription](#) (const char **atts)
- void [HandleModule](#) (const char **atts)
- void [HandleModuleEntry](#) (const char **atts)
- void [HandleModuleEntryDescription](#) (const char **atts)
- void [HandleModuleInclude](#) (const char **atts)
- int [Read](#) ()
- void [SetFilename](#) (const char *filename)

Protected Member Functions

- void [HandleDescription](#) (const char **atts)
- void [HandleEntry](#) (const char **atts)

12.406.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

Note

bla Will read the Private.xml file

12.406.2 Constructor & Destructor Documentation

12.406.2.1 XMLPrivateDictReader()

`gdcm::XMLPrivateDictReader::XMLPrivateDictReader ()`

12.406.2.2 ~XMLPrivateDictReader()

`gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ()` [inline]

12.406.3 Member Function Documentation

12.406.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (  
    const char * data,  
    int length) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

12.406.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (  
    const char * name) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

12.406.3.3 GetPrivateDict()

```
const PrivateDict & gdcm::XMLPrivateDictReader::GetPrivateDict () [inline]
```

12.406.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (  
    const char ** atts) [protected]
```

12.406.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (  
    const char ** atts) [protected]
```

12.406.3.6 StartElement()

```
void gdcm::XMLPrivateDictReader::StartElement (  
    const char * name,  
    const char ** atts) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

The documentation for this class was generated from the following file:

- [gdcmXMLPrivateDictReader.h](#)

Chapter 13

File Documentation

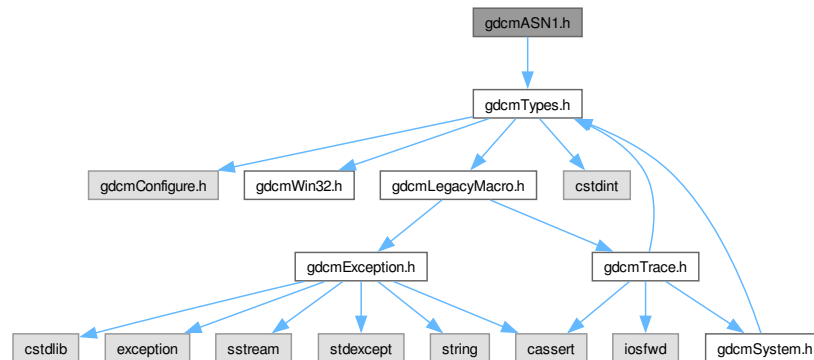
13.1 README.txt File Reference

13.2 TestsList.txt File Reference

13.3 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class `gdcm::ASN1`
Class for `ASN1`.

Namespaces

- namespace [gdcm](#)

13.4 gdcmASN1.h

[Go to the documentation of this file.](#)

```

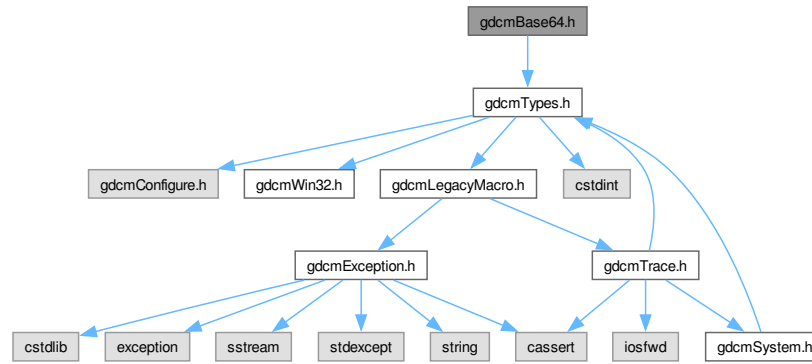
00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMASN1_H
00015  #define GDCMASN1_H
00016  00017  #include "gdcmTypes.h"
00018  00019  00020  namespace gdcm
00021  {
00022  //-----
00023  class ASN1Internals;
00028  class GDCM_EXPORT ASN1
00029  {
00030  public :
00031    ASN1();
00032    ~ASN1();
00033  00034    static bool ParseDumpFile(const char *filename);
00035  00036    static bool ParseDump(const char *array, size_t length);
00037  00038    ASN1(const ASN1&) = delete;
00039    void operator=(const ASN1&) = delete;
00040  protected:
00041    int TestPBKDF2();
00042  00043  private:
00044    ASN1Internals *Internals;
00045  };
00046  } // end namespace gdcm
00047  //-----
00048  #endif //GDCMASN1_H

```

13.5 gdcmBase64.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmBase64.h:



Classes

- class [gdcm::Base64](#)
Class for [Base64](#).

Namespaces

- namespace [gdcm](#)

13.6 gdcmBase64.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBASE64_H
00015  #define GDCMBASE64_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {

```

```

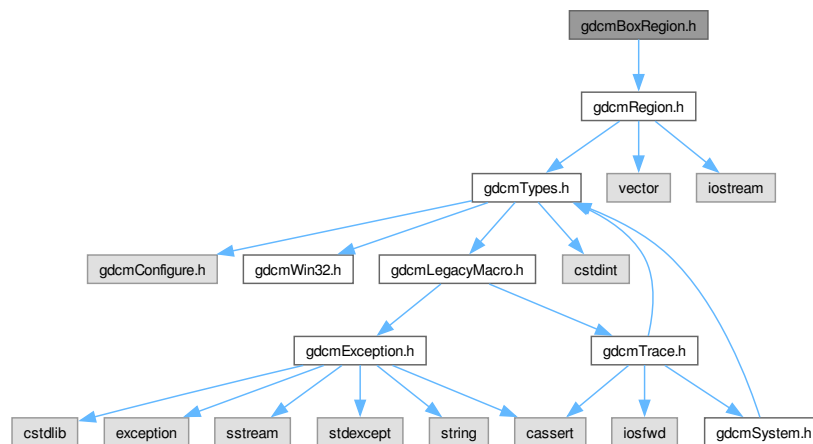
00025 class GDCM_EXPORT Base64
00026 {
00027 public:
00028
00032 static size_t GetEncodeLength(const char *src, size_t srclen );
00033
00045 static size_t Encode( char *dst, size_t dlen, const char *src, size_t slen );
00046
00050 static size_t GetDecodeLength( const char *src, size_t len );
00051
00062 static size_t Decode( char *dst, size_t dlen, const char *src, size_t slen );
00063
00064 Base64(const Base64&) = delete;
00065 void operator=(const Base64&) = delete;
00066 };
00067
00068 } // end namespace gdcmm
00069
00070 #endif // GDCMBASE64_H

```

13.7 gdcmmBoxRegion.h File Reference

#include "gdcmmRegion.h"

Include dependency graph for gdcmmBoxRegion.h:



Classes

- class [gdcmm::BoxRegion](#)
Class for manipulation box region.

Namespaces

- namespace [gdcmm](#)

13.8 gdcmBoxRegion.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMBOXREGION_H
00015 #define GDCMBOXREGION_H
00016
00017 #include "gdcmRegion.h"
00018
00019 namespace gdcm
00020 {
00021  class BoxRegionInternals;
00022  //-----
00023  class GDCM_EXPORT BoxRegion : public Region
00024  {
00025  public :
00026    BoxRegion();
00027    ~BoxRegion() override;
00028
00029    void SetDomain(unsigned int xmin, unsigned int xmax,
00030                  unsigned int ymin, unsigned int ymax,
00031                  unsigned int zmin, unsigned int zmax);
00032
00033    unsigned int GetXMin() const;
00034    unsigned int GetXMax() const;
00035    unsigned int GetYMin() const;
00036    unsigned int GetYMax() const;
00037    unsigned int GetZMin() const;
00038    unsigned int GetZMax() const;
00039
00040    // Satisfy pure virtual parent class
00041    Region *Clone() const override;
00042    bool Empty() const override;
00043    bool IsValid() const override;
00044    size_t Area() const override;
00045    BoxRegion ComputeBoundingBox() override;
00046
00047    void Print(std::ostream &os = std::cout) const override;
00048
00049    static BoxRegion BoundingBox(BoxRegion const & b1, BoxRegion const & b2 );
00050
00051    BoxRegion(const BoxRegion&);
00052    void operator=(const BoxRegion&);
00053  private:
00054    BoxRegionInternals *Internals;
00055  };
00056
00057 } // end namespace gdcm
00058 //-----
00059 #endif //GDCMREGION_H

```

13.9 gdcmByteSwap.h File Reference

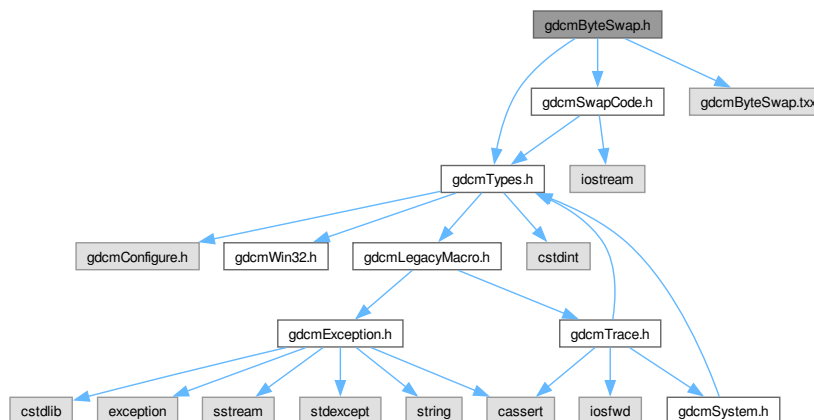
```

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"

```

```
#include "gdcmByteSwap.txx"
```

Include dependency graph for gdcmByteSwap.h:



Classes

- class `gdcm::ByteSwap< T >`
`ByteSwap`.

Namespaces

- namespace `gdcm`

13.10 gdcmByteSwap.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBYTESWAP_H
00015  #define GDCMBYTESWAP_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmSwapCode.h"
00019
00020  namespace gdcm
00021  {

```

```

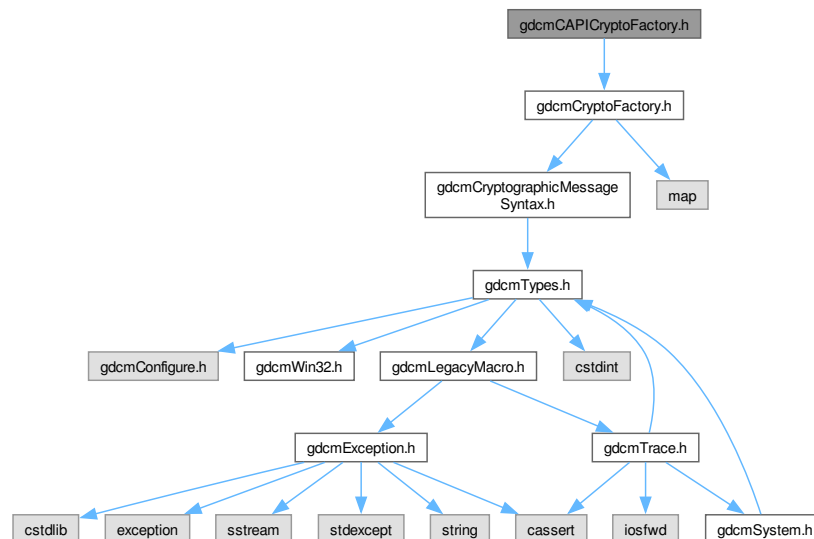
00022
00029 template<class T>
00030 class ByteSwap
00031 {
00032 public:
00034     static bool SystemIsBigEndian ();
00035     static bool SystemIsLittleEndian ();
00036
00037     static void Swap(T &p);
00038     static void SwapFromSwapCodeIntoSystem(T &p, SwapCode const &sc);
00039     static void SwapRange(T *p, unsigned int num);
00040     static void SwapRangeFromSwapCodeIntoSystem(T *p, SwapCode const &sc,
00041         std::streamoff num);
00042
00043 protected:
00044     // ByteSwap() {}
00045     // ~ByteSwap() {}
00046
00047 private:
00048
00049 };
00050
00055
00056 } // end namespace gdcm
00057
00058 #include "gdcmByteSwap.txx"
00059
00060 #endif //GDCMBYTESWAP_H

```

13.11 gdcmCAPICryptoFactory.h File Reference

#include "gdcmCryptoFactory.h"

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class `gdcm::CAPICryptoFactory`

Namespaces

- namespace [gdcm](#)

13.12 gdcmCAPICryptoFactory.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMCAPICRYPTOFACTORY_H
00015  #define GDCMCAPICRYPTOFACTORY_H
00016  00017  #include "gdcmCryptoFactory.h"
00018  00019  namespace gdcm
00020  {
00021  00022  class GDCM_EXPORT CAPICryptoFactory : public CryptoFactory
00023  {
00024  public:
00025    CAPICryptoFactory(CryptoLib id);
00026    CryptographicMessageSyntax* CreateCMSProvider();
00027  00028  private:
00029    CAPICryptoFactory() {}
00030  };
00031  00032  } // end namespace gdcm
00033  00034  #endif //GDCMCAPICRYPTOFACTORY_H

```

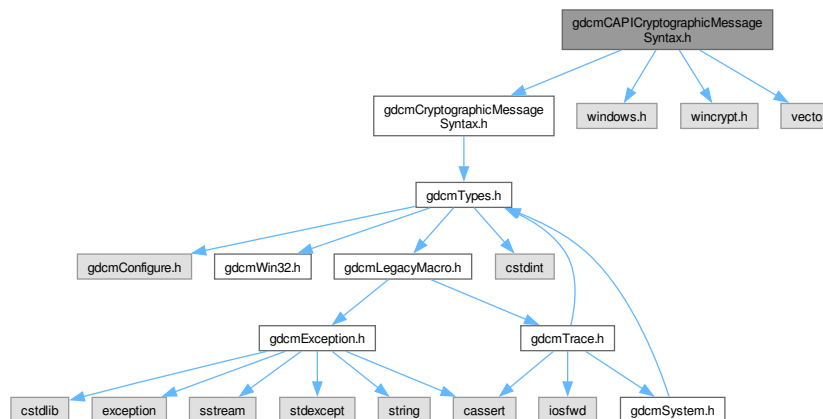
13.13 gdcmCAPICryptographicMessageSyntax.h File Reference

```

#include "gdcmCryptographicMessageSyntax.h"
#include <windows.h>
#include <wincrypt.h>
#include <vector>

```

Include dependency graph for gdcmCAPICryptographicMessageSyntax.h:



Classes

- class [gdcm::CAPICryptographicMessageSyntax](#)

Namespaces

- namespace [gdcm](#)

13.14 gdcmCAPICryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcmCryptographicMessageSyntax.h"
00018 #include <windows.h>
00019 #include <wincrypt.h>
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT CAPICryptographicMessageSyntax : public CryptographicMessageSyntax

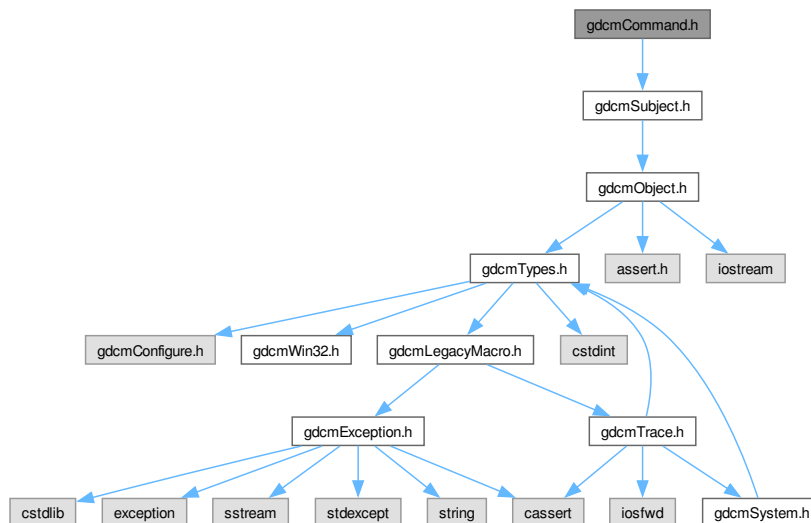
```

```
00026 {
00027 public:
00028     CAPICryptographicMessageSyntax();
00029     ~CAPICryptographicMessageSyntax();
00030
00031     // X.509
00032     bool ParseCertificateFile( const char *filename );
00033     bool ParseKeyFile( const char *filename );
00034
00035     // PBE
00036     bool SetPassword(const char * pass, size_t passLen);
00037
00038     void SetCipherType(CipherTypes type);
00039
00040     CipherTypes GetCipherType() const;
00041
00042     bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00043     bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00044
00045     bool GetInitialized() const
00046     {
00047         return initialized;
00048     }
00049
00050 private:
00051     bool Initialize();
00052     static ALG_ID GetAlgIdByObjId(const char * pszObjId);
00053     static const char *GetCipherObjId() const;
00054     static void ReverseBytes(unsigned char* data, DWORD len);
00055     static bool LoadFile(const char * filename, unsigned char* & buffer, DWORD & bufLen);
00056
00057 private:
00058     bool initialized;
00059     HCRYPTPROV hProv;
00060     std::vector<PCCERT_CONTEXT> certifList;
00061     HCRYPTKEY hRsaPrivK;
00062     CipherTypes cipherType;
00063 };
00064
00065 } // end namespace gdcms
00066
00067 #endif // GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
```

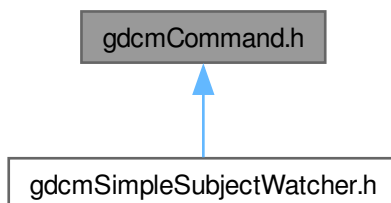
13.15 gdcCommand.h File Reference

#include "gdcSubject.h"

Include dependency graph for gdcCommand.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdc::Command](#)
[Command](#) superclass for callback/observer methods.
- class [gdc::MemberCommand< T >](#)
[Command](#) subclass that calls a pointer to a member function.
- class [gdc::SimpleMemberCommand< T >](#)
[Command](#) subclass that calls a pointer to a member function.

Namespaces

- namespace [gdcm](#)

13.16 gdcmCommand.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCOMMAND_H
00015 #define GDCMCOMMAND_H
00016
00017 #include "gdcmSubject.h"
00018
00019 namespace gdcm
00020 {
00021   class Event;
00022
00023   class GDCM_EXPORT Command : public Subject
00024   {
00025   public :
00026     Command(const Command&) = delete;
00027     void operator=(const Command&) = delete;
00028
00029     virtual void Execute(Subject *caller, const Event & event ) = 0;
00030
00031     virtual void Execute(const Subject *caller, const Event & event ) = 0;
00032
00033   protected:
00034     Command();
00035     ~Command() override;
00036   };
00037
00038   template <class T>
00039   class MemberCommand : public Command
00040   {
00041   public:
00042     typedef void (T::*TMemberFunctionPointer)(Subject*, const Event &);
00043     typedef void (T::*TConstMemberFunctionPointer)(const Subject*,
00044                                                     const Event &);
00045
00046     typedef MemberCommand    Self;
00047     //typedef SmartPointer<Self>  Pointer;
00048
00049     MemberCommand(const Self&) = delete;
00050     void operator=(const Self&) = delete;
00051
00052     static SmartPointer<MemberCommand> New()
00053     {
00054       return new MemberCommand;
00055     }
00056
00057     //gdcmTypeMacro(MemberCommand,Command);
00058
00059     void SetCallbackFunction(T* object,
00060                             TMemberFunctionPointer memberFunction)
00061     {
00062       m_This = object;
00063       m_MemberFunction = memberFunction;
00064     }

```



```

00088 void SetCallbackFunction(T* object,
00089                          TConstMemberFunctionPointer memberFunction)
00090 {
00091     m_This = object;
00092     m_ConstMemberFunction = memberFunction;
00093 }
00094
00096 void Execute(Subject *caller, const Event & event ) override
00097 {
00098     if( m_MemberFunction )
00099     {
00100         ((*m_This).*(m_MemberFunction))(caller, event);
00101     }
00102 }
00103
00105 void Execute( const Subject *caller, const Event & event ) override
00106 {
00107     if( m_ConstMemberFunction )
00108     {
00109         ((*m_This).*(m_ConstMemberFunction))(caller, event);
00110     }
00111 }
00112
00113 protected:
00114     T* m_This;
00115     TMemberFunctionPointer m_MemberFunction;
00116     TConstMemberFunctionPointer m_ConstMemberFunction;
00117     MemberCommand():m_This(nullptr),m_MemberFunction(nullptr),m_ConstMemberFunction(nullptr) {}
00118     ~MemberCommand() override= default;
00119 };
00120
00122
00129 template <typename T>
00130 class SimpleMemberCommand : public Command
00131 {
00132 public:
00133     typedef void (T::*TMemberFunctionPointer)();
00134     typedef SimpleMemberCommand Self;
00135     //typedef SmartPointer<Self> Pointer;
00136     SimpleMemberCommand(const Self&) = delete;
00137     void operator=(const Self&) = delete;
00138     //gdcmTypeMacro(SimpleMemberCommand,Command);
00139
00140     static SmartPointer<SimpleMemberCommand> New()
00141     {
00142         return new SimpleMemberCommand;
00143     }
00144
00145     void SetCallbackFunction(T* object,
00146                             TMemberFunctionPointer memberFunction)
00147     {
00148         m_This = object;
00149         m_MemberFunction = memberFunction;
00150     }
00151
00152     void Execute(Subject *,const Event & ) override
00153     {
00154         if( m_MemberFunction )
00155         {
00156             ((*m_This).*(m_MemberFunction))();
00157         }
00158     }
00159
00160     void Execute(const Subject *,const Event & ) override
00161     {
00162         if( m_MemberFunction )
00163         {
00164             ((*m_This).*(m_MemberFunction))();
00165         }
00166     }
00167
00168     protected:
00169         T* m_This;
00170         TMemberFunctionPointer m_MemberFunction;
00171         SimpleMemberCommand():m_This(nullptr),m_MemberFunction(nullptr) {}
00172         ~SimpleMemberCommand() override= default;
00173 };

```

```

00183
00184 } // end namespace gdcM
00185 //-----
00186 #endif //GDCMCOMMAND_H

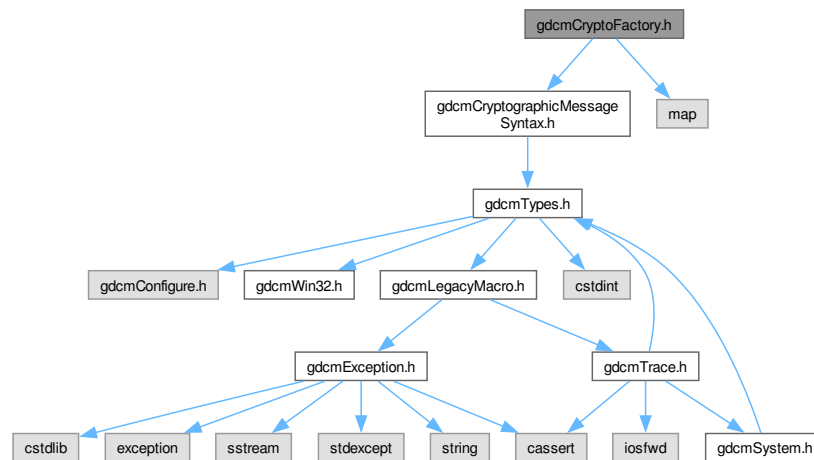
```

13.17 gdcmCryptoFactory.h File Reference

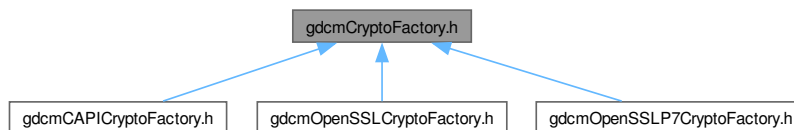
```
#include "gdcmCryptographicMessageSyntax.h"
```

```
#include <map>
```

Include dependency graph for gdcmCryptoFactory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcM::CryptoFactory](#)
Class to do handle the crypto factory.

Namespaces

- namespace [gdcM](#)

13.18 gdcmCryptoFactory.h

[Go to the documentation of this file.](#)

```

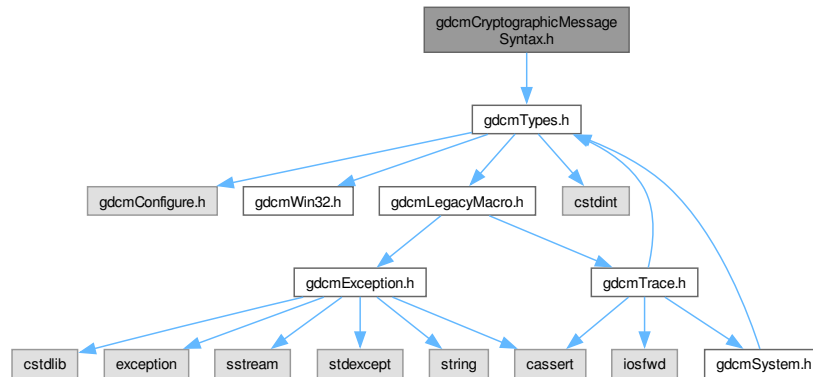
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCRYPTOFACTORY_H
00015 #define GDCMCRYPTOFACTORY_H
00016
00017 #include "gdcmCryptographicMessageSyntax.h"
00018 #include <map>
00019
00020 namespace gdcm
00021 {
00022
00035 class GDCM_EXPORT CryptoFactory
00036 {
00037 public:
00038     enum CryptoLib {DEFAULT = 0, OPENSSSL = 1, CAPI = 2, OPENSSSLP7 = 3};
00039
00040     virtual CryptographicMessageSyntax* CreateCMSProvider() = 0;
00041     static CryptoFactory* GetFactoryInstance(CryptoLib id = DEFAULT);
00042
00043 protected:
00044     CryptoFactory(CryptoLib id)
00045     {
00046         AddLib(id, this);
00047     }
00048
00049 private:
00050     static std::map<CryptoLib, CryptoFactory*> getInstanceMap()
00051     {
00052         static std::map<CryptoLib, CryptoFactory*> libs;
00053         return libs;
00054     }
00055
00056     static void AddLib(CryptoLib id, CryptoFactory* f)
00057     {
00058         if (getInstanceMap().insert(std::pair<CryptoLib, CryptoFactory*>(id, f)).second == false)
00059         {
00060             gdcmErrorMacro( "Library already registered under id " « (int)id );
00061         }
00062     }
00063
00064 protected:
00065     CryptoFactory()= default;
00066     ~CryptoFactory()= default;
00067 };
00068
00069 } // end namespace gdcm
00070
00071 #endif // GDCMCRYPTOFACTORY_H

```

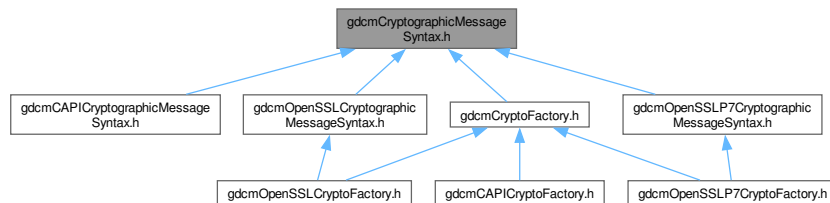
13.19 gdcmCryptographicMessageSyntax.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::CryptographicMessageSyntax](#)

Namespaces

- namespace [gdcm](#)

13.20 gdcmCryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

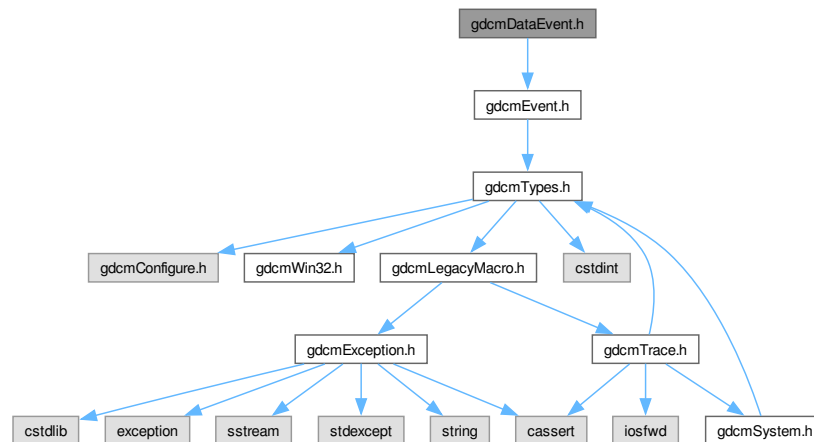
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class GDCM_EXPORT CryptographicMessageSyntax
00023 {
00024 public:
00025     CryptographicMessageSyntax() = default;
00026
00027     virtual ~CryptographicMessageSyntax() = default;
00028     CryptographicMessageSyntax(const CryptographicMessageSyntax&) = delete;
00029     void operator=(const CryptographicMessageSyntax&) = delete;
00030
00031     typedef enum {
00032         DES3_CIPHER, // Triple DES
00033         AES128_CIPHER, // CBC AES
00034         AES192_CIPHER, // ' '
00035         AES256_CIPHER // ' '
00036     } CipherTypes;
00037
00038     // X.509
00039     virtual bool ParseCertificateFile( const char *filename ) = 0;
00040     virtual bool ParseKeyFile( const char *filename ) = 0;
00041
00042     // PBE
00043     virtual bool SetPassword(const char * pass, size_t passLen) = 0;
00044
00046     virtual bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
00048     virtual bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
00049
00050     virtual void SetCipherType(CipherTypes type) = 0;
00051
00052     virtual CipherTypes GetCipherType() const = 0;
00053 };
00054
00055 } // end namespace gdcm
00056
00057 #endif //GDCMCRYPTOGRAPHICMESSAGESYNTAX_H

```

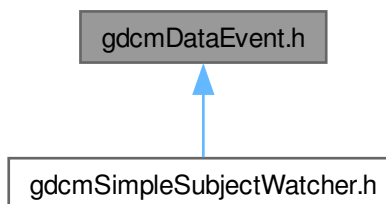
13.21 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::DataEvent`
`DataEvent`.

Namespaces

- namespace `gdcm`

13.22 gdcmDataEvent.h

[Go to the documentation of this file.](#)

```

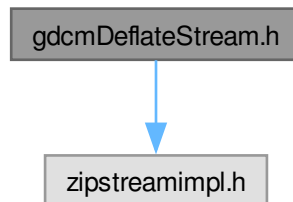
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDATAEVENT_H
00015 #define GDCMDATAEVENT_H
00016
00017 #include "gdcmEvent.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class DataEvent : public AnyEvent
00023 {
00024 public:
00025     typedef DataEvent Self;
00026     typedef AnyEvent Superclass;
00027     DataEvent(const char *bytes = nullptr, size_t len = 0):Bytes(bytes),Length(len) {}
00028     ~DataEvent() override = default;
00029     DataEvent(const Self&s) : AnyEvent(s), Bytes(nullptr), Length(0) {}
00030     void operator=(const Self&) = delete;
00031
00032     const char * GetEventName() const override { return "DataEvent"; }
00033     bool CheckEvent(const ::gdcm::Event* e) const override
00034     { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
00035     ::gdcm::Event* MakeObject() const override
00036     { return new Self; }
00037
00038     void SetData(const char *bytes, size_t len) {
00039         Bytes = bytes;
00040         Length = len;
00041     }
00042     size_t GetDataLength() const { return Length; }
00043     const char *GetData() const { return Bytes; }
00044
00045     //std::string GetValueAsString() const { return; }
00046
00047 private:
00048     const char *Bytes;
00049     size_t Length;
00050 };
00051
00052 } // end namespace gdcm
00053
00054 #endif //GDCMDATAEVENT_H

```

13.23 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:



13.24 gdcmDeflateStream.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDEFLATESTREAM_H
00015 #define GDCMDEFLATESTREAM_H
00016
00017 #include "zipstreamimpl.h"
00018
00019 #endif //GDCMDEFLATESTREAM_H
  
```

13.25 gdcmDirectory.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

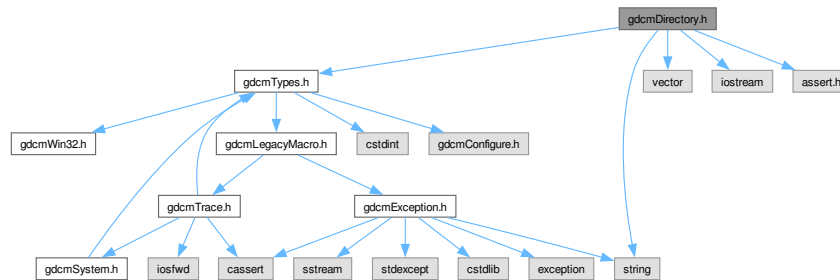
```
#include <vector>
```

```
#include <iostream>
```



```
#include <assert.h>
```

Include dependency graph for gdcmDirectory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Directory](#)
Class for manipulation directories.

Namespaces

- namespace [gdcm](#)

Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [Directory](#) &d)

13.26 gdcmDirectory.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
  
```

```

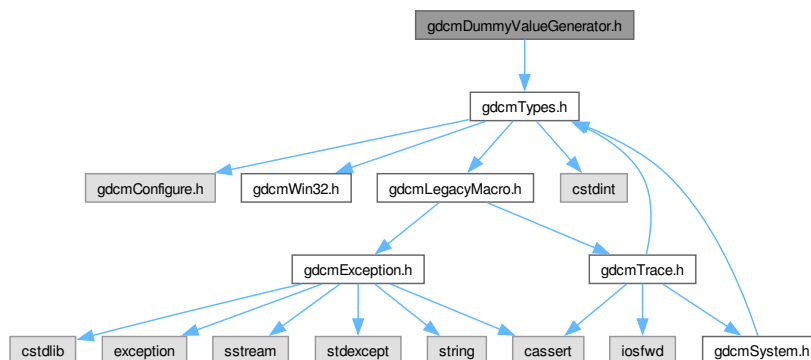
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00014 #ifndef GDCMDIRECTORY_H
00015 #define GDCMDIRECTORY_H
00016
00017 #include "gdcTypes.h"
00018
00019 #include <string>
00020 #include <vector>
00021 #include <iostream>
00022 #include <assert.h>
00023
00024 namespace gdc
00025 {
00041 //-----
00042 class GDCM_EXPORT Directory
00043 {
00044     friend std::ostream& operator<<(std::ostream &_os, const Directory &d);
00045 public :
00046     Directory() = default;
00047     ~Directory() = default;
00048     typedef std::string FilenameType;
00049     typedef std::vector<FilenameType> FilenamesType;
00050
00052     void Print(std::ostream &os = std::cout) const;
00053
00055     FilenameType const &GetToplevel() const { return Toplevel; }
00056
00058     FilenamesType const &GetFilenames() const {
00059         gdc_assert( !(Toplevel.empty()) && "Need to call Explore first" );
00060         return Filenames; }
00061
00063     FilenamesType const &GetDirectories() const { return Directories; }
00064
00067     unsigned int Load(FilenameType const &name, bool recursive = false);
00068
00069     // \todo later: GLOB
00070     // The glob() function searches for all the pathnames matching pattern according to
00071     // the rules used by the shell (see glob(7)). No tilde expansion or parameter
00072     // substitution is done; if you want these, use wordexp(3).
00073     // int Glob(...);
00074
00075 protected:
00077     unsigned int Explore(FilenameType const &name, bool recursive);
00078
00079 private :
00081     FilenamesType Filenames;
00082     FilenamesType Directories;
00083
00085     FilenameType Toplevel;
00086 };
00087 //-----
00088 inline std::ostream& operator<<(std::ostream &os, const Directory &d)
00089 {
00090     d.Print( os );
00091     return os;
00092 }
00093
00094 } // end namespace gdc
00095 //-----
00096 #endif //GDCMDIRECTORY_H

```

13.27 gdcmDummyValueGenerator.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class [gdcm::DummyValueGenerator](#)
Class for generating dummy value.

Namespaces

- namespace [gdcm](#)

13.28 gdcmDummyValueGenerator.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDUMMYVALUEGENERATOR_H
00015  #define GDCMDUMMYVALUEGENERATOR_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {

```

```

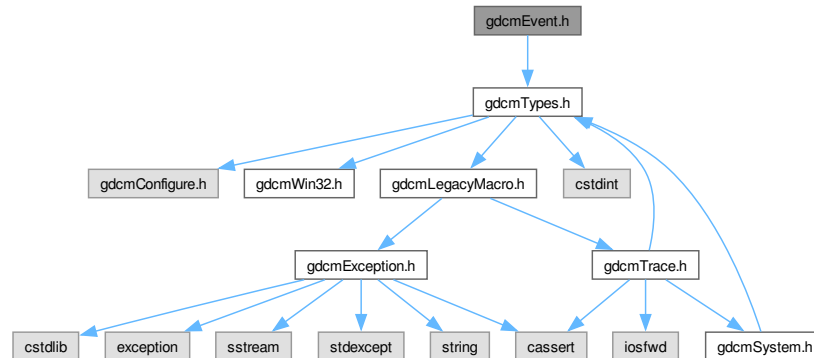
00021
00026 class GDCM_EXPORT DummyValueGenerator
00027 {
00028 public:
00029
00035 static const char* Generate(const char *input);
00036
00037 private:
00038 };
00039
00040
00041 } // end namespace gdcM
00042
00043 #endif //GDCMDUMMYVALUEGENERATOR_H

```

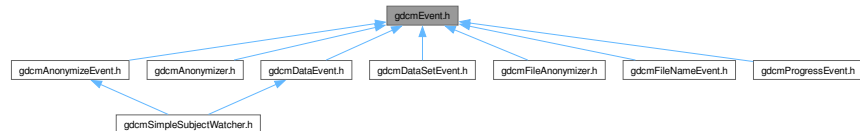
13.29 gdcMEvent.h File Reference

#include "gdcMTypes.h"

Include dependency graph for gdcMEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcM::AbortEvent](#)
- class [gdcM::AnyEvent](#)
- class [gdcM::EndEvent](#)
- class [gdcM::Event](#)

superclass for callback/observer methods

- class [gdcm::ExitEvent](#)
- class [gdcm::InitializeEvent](#)
- class [gdcm::IterationEvent](#)
- class [gdcm::ModifiedEvent](#)
- class [gdcm::NoEvent](#)
- class [gdcm::StartEvent](#)
- class [gdcm::UserEvent](#)

Namespaces

- namespace [gdcm](#)

Macros

- `#define gdcmEventMacro(classname, super)`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Event &e)`
Generic inserter operator for [Event](#) and its subclasses.

13.29.1 Macro Definition Documentation

13.29.1.1 gdcmEventMacro

```
#define gdcmEventMacro(
    classname,
    super)
```

Value:

```
\
class classname : public super { \
public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() override = default; \
    virtual const char * GetEventName() const override { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const override \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const override \
    { return new Self; } \
    classname(const Self&s) : super(s){} \
private: \
    void operator=(const Self&); \
}
```

13.30 gdcmEvent.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMEVENT_H
00015 #define GDCMEVENT_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021  //-----
00022  class GDCM_EXPORT Event
00023  {
00024  public :
00025      Event();
00026      virtual ~Event();
00027      Event(const Event&);
00028      void operator=(const Event&) = delete;
00029
00030      virtual Event* MakeObject() const = 0;
00031
00032      virtual void Print(std::ostream& os) const;
00033
00034      virtual const char * GetEventName() const = 0;
00035
00036      virtual bool CheckEvent(const Event*) const = 0;
00037  };
00038
00039 inline std::ostream& operator<<(std::ostream& os, const Event &e)
00040 {
00041     e.Print(os);
00042     return os;
00043 }
00044
00045 /*
00046  * Macro for creating new Events
00047  */
00048 #define gdcmEventMacro( classname , super ) \
00049 \
00050 class classname : public super { \
00051 public: \
00052     typedef classname Self; \
00053     typedef super Superclass; \
00054     classname() {} \
00055     virtual ~classname() override = default; \
00056     virtual const char * GetEventName() const override { return #classname; } \
00057     virtual bool CheckEvent(const ::gdcm::Event* e) const override \
00058     { return dynamic_cast<const Self*>(e) ? true : false; } \
00059     virtual ::gdcm::Event* MakeObject() const override \
00060     { return new Self; } \
00061     classname(const Self&s) : super(s){} \
00062 private: \
00063     void operator=(const Self&); \
00064 }
00065
00066 gdcmEventMacro( NoEvent , Event );
00067 gdcmEventMacro( AnyEvent , Event );
00068 gdcmEventMacro( StartEvent , AnyEvent );
00069 gdcmEventMacro( EndEvent , AnyEvent );
00070 //gdcmEventMacro( ProgressEvent , AnyEvent );
00071 gdcmEventMacro( ExitEvent , AnyEvent );
00072 gdcmEventMacro( AbortEvent , AnyEvent );

```

```

00089 gdcmEventMacro( ModifiedEvent      , AnyEvent );
00090 gdcmEventMacro( InitializeEvent    , AnyEvent );
00091 gdcmEventMacro( IterationEvent     , AnyEvent );
00092 //gdcmEventMacro( AnonymizeEvent    , AnyEvent );
00093 gdcmEventMacro( UserEvent          , AnyEvent );
00094
00095
00096 } // end namespace gdcm
00097 //-----
00098 #endif //GDCMEVENT_H

```

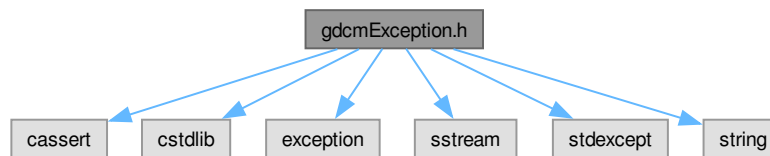
13.31 gdcmException.h File Reference

```

#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>

```

Include dependency graph for gdcmException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Exception](#)
[Exception](#).

Namespaces

- namespace [gdcm](#)

Macros

- `#define` [gdcm_assert](#)(cond)
- `#define` [gdcm_debug_assert](#)(cond)
- `#define` [gdcm_forced_assert](#)(cond)

13.31.1 Macro Definition Documentation

13.31.1.1 gdcm_assert

```
#define gdcm_assert(  
    cond)
```

Value:

```
if (!(cond)) throw gdcm::Exception("An invalid logic behavior occurred" #cond, __FILE__ , __LINE__)
```

Examples

CStoreQtProgress.cxx, CheckBigEndianBug.cxx, DiscriminateVolume.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, ELSCINT1WaveToText.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_ GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndPrintAttributes.cxx, StreamImageReaderTest.cxx, VolumeSorter.cxx, and pmsct_rgb1.cxx.

Referenced by `gdcm::network::AAssociateRQPDU::AAssociateRQPDU()`, `gdcm::CSAHeaderDict::CSAHeaderDict()`, `gdcm::Dict::Dict()`, `gdcm::LookupTable::LookupTable()`, `gdcm::CSAHeaderDict::AddCSAHeaderDictEntry()`, `gdcm::Dict::AddDictEntry()`, `gdcm::PrivateDict::AddDictEntry()`, `gdcm::Macros::AddMacro()`, `gdcm::Modules::AddModule()`, `gdcm::network::PDataTFPDU::AddPresentationDataValue()`, `gdcm::Clamp()`, `gdcm::DataSet::Clear()`, `gdcm::DataSet::ComputeGroupLength()`, `gdcm::ByteBuffer::Get()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetA`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM`, `gdcm::Element< TVR, TVM >::GetAsDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::PixelFormat::GetBitsStored()`, `gdcm::Pixmap::GetCurve()`, `gdcm::Pixmap::GetCurve()`, `gdcm::Dict::GetDictEntry()`, `gdcm::PrivateDict::GetDictEntry()`, `gdcm::Dict::GetDictEntryByKeyword()`, `gdcm::Dict::GetDictEntryByName()`, `gdcm::Directory::GetFileNames()`, `gdcm::FilenameGenerator::GetFileNames()`, `gdcm::PixelFormat::GetHighBit()`, `gdcm::IODs::GetIOD()`, `gdcm::Dict::GetKeywordFromTag()`, `gdcm::DataSet::GetLength()`, `gdcm::Macros::GetMacro()`, `gdcm::network::PresentationDataValue::GetMessageHeader()`, `gdcm::Modules::GetModule()`, `gdcm::Pixmap::GetOverlay()`, `gdcm::Pixmap::GetOverlay()`, `gdcm::network::AAssociateRQPDU::GetPresentationContext()`, `gdcm::network::AAssociateAC`, `gdcm::network::PDataTFPDU::GetPresentationDataValue()`, `gdcm::VR::GetSize()`, `gdcm::Table::GetTableEntry()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetValue`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1`, `gdcm::Element< TVR, TVM >::GetValue()`, `gdcm::Element< TVR, TVM >::GetValue()`, `gdcm::Element< TVR, VM::VM1`, `gdcm::DataSet::InsertDataElement()`, `gdcm::Item::InsertDataElement()`, `gdcm::Table::InsertEntry()`, `gdcm::ByteValue::IsEmpty()`, `gdcm::ByteValue::IsPrintable()`, `gdcm::Scanner2::ltstr::operator()`, `gdcm::Scanner::ltstr::operat`, `gdcm::StrictScanner2::ltstr::operator()`, `gdcm::StrictScanner::ltstr::operator()`, `gdcm::SmartPointer< Value >::operator*(`, `gdcm::BasicOffsetTable::operator<<`, `gdcm::CSAElement::operator<<`, `gdcm::File::operator<<`, `gdcm::VM::operator<<`, `gdcm::network::PresentationContextRQ::operator==()`, `gdcm::PresentationContext::operator==()`, `gdcm::Tag::operator[]()`, `gdcm::Tag::operator[]()`, `gdcm::ApplicationEntity::Print()`, `gdcm::Element< TVR, VM::VM1_n >::Print()`, `gdcm::SequenceOfFragments::Print()`, `gdcm::ByteValue::PrintGroupLength()`, `gdcm::BasicOffsetTable::Read()`, `gdcm::ByteValue::Read()`, `gdcm::EncodingImplementation< VR::VRASCII >::Read()`, `gdcm::EncodingImplementation< VR`, `gdcm::Item::Read()`, `gdcm::SequenceOfFragments::Read()`, `gdcm::SequenceOfItems::Read()`, `gdcm::VR::Read()`, `gdcm::VL::Read16()`, `gdcm::Fragment::ReadBacktrack()`, `gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeL`, `gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength()`, `gdcm::EncodingImplementation< VR::VRBI`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB()`, `gdcm::Object::Register()`, `gdcm::DataSet::Remove()`, `gdcm::PrivateDict::RemoveDictEntry()`, `gdcm::Pixmap::RemoveOverlay`, `gdcm::ImageChangePhotometricInterpretation::RGB2YBR()`, `gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPI`, `gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels()`, `gdcm::SurfaceHelper::RGBToRecommendedDisplayCIE`, `gdcm::SurfaceHelper::RGBToRecommendedDisplayGrayscale()`, `gdcm::Element< TVR, VM::VM1_n >::Set()`, `gdcm::Element< TVR, VM::VM1_n >::SetArray()`, `gdcm::ApplicationEntity::SetBlob()`, `gdcm::Attribute< Group, Element,`


```

gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(),
gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(),
gdcm::PersonName::SetComponents(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(),
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetLength(),
gdcm::Element< TVR, VM::VM1_n >::SetLength(), gdcm::network::PresentationDataValue::SetMessageHeader(),
gdcm::Element< TVR, TVM >::SetNoSwap(), gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(),
gdcm::ImageCodec::SetPlanarConfiguration(), gdcm::network::PresentationDataValue::SetPresentationContextID(),
gdcm::Tag::SetPrivateCreator(), gdcm::PixelFormat::SetSamplesPerPixel(), gdcm::Attribute< Group, Element, TVR, TVM >::SetTag(),
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcm::Element< TVR, TVM >::SetValue(),
gdcm::Element< TVR, VM::VM1_n >::SetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValues(),
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues(), gdcm::GroupDict::Size(), gdcm::Object::UnRegister(),
gdcm::ByteValue::Write(), gdcm::EncodingImplementation< VR::VRASCII >::Write(), gdcm::EncodingImplementation< VR::VRASCII >::Write(),
gdcm::EncodingImplementation< VR::VRBINARY >::Write(), gdcm::Fragment::Write(), gdcm::Item::Write(),
gdcm::SequenceOfFragments::Write(), gdcm::VR::Write(), gdcm::VL::Write16(), gdcm::ByteValue::WriteBuffer(),
and gdcm::ImageChangePhotometricInterpretation::YBR2RGB().

```

13.31.1.2 gdcm_debug_assert

```

#define gdcm_debug_assert(
    cond)

```

Value:

```
gdcm_assert(cond)
```

13.31.1.3 gdcm_forced_assert

```

#define gdcm_forced_assert(
    cond)

```

Value:

```
assert(cond)
```

Referenced by [gdcm::Object::~~Object\(\)](#).

13.32 gdcmException.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMEXCEPTION_H
00015 #define GDCMEXCEPTION_H

```

```

00016
00017 #include <cassert>
00018 #include <cstdlib> // NULL
00019 #include <exception>
00020 #include <sstream> // ostream
00021 #include <stdexcept> // logic_error
00022 #include <string>
00023
00024 // Disable clang warning "dynamic exception specifications are deprecated".
00025 // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
00026 // specifier we'll get an error in C++03 by not matching the superclass.
00027 #if defined(__clang__) && defined(__has_warning)
00028 # if __has_warning("-Wdeprecated")
00029 # pragma clang diagnostic push
00030 # pragma clang diagnostic ignored "-Wdeprecated"
00031 # endif
00032 #endif
00033
00034 #define gdcmm_forced_assert(cond) assert(cond)
00035
00036 namespace gdcmm
00037 {
00038
00039 class Exception : public std::exception
00040 {
00041     typedef std::logic_error StringHolder;
00042
00043     static StringHolder CreateWhat(const char* const desc,
00044                                   const char* const file,
00045                                   const unsigned int lineNumber,
00046                                   const char* const func)
00047     {
00048         gdcmm_forced_assert(desc != nullptr);
00049         gdcmm_forced_assert(file != nullptr);
00050         gdcmm_forced_assert(func != nullptr);
00051         std::ostringstream oswhat;
00052         oswhat << file << ":" << lineNumber << " (" << func << "):\n";
00053         oswhat << desc;
00054         return StringHolder( oswhat.str() );
00055     }
00056
00057 public:
00058     explicit Exception(const char *desc = "None",
00059                      const char *file = __FILE__,
00060                      unsigned int lineNumber = __LINE__,
00061                      // FIXME: __PRETTY_FUNCTION__ is the non-mangled version for __GNUC__ compiler
00062                      const char *func = "" /* __FUNCTION__ */)
00063     :
00064     What( CreateWhat(desc, file, lineNumber, func) ),
00065     Description(desc)
00066     {
00067     }
00068
00069 ~Exception() throw() override {}
00070
00071 const char* what() const throw() override
00072 {
00073     return What.what();
00074 }
00075
00076 const char * GetDescription() const { return Description.what(); }
00077
00078 private:
00079     StringHolder What;
00080     StringHolder Description;
00081 };
00082
00083 } // end namespace gdcmm
00084
00085 // Always defined
00086 #define gdcmm_assert(cond) \
00087     if (!(cond)) throw gdcmm::Exception("An invalid logic behavior occurred" #cond, __FILE__, __LINE__)
00088
00089 /* Asserts that should only exist in debug builds. */
00090 #ifndef NDEBUG // checks in debug builds and elision in release builds (like assert)
00091 #define gdcmm_debug_assert(cond) gdcmm_assert(cond)
00092 #else
00093 #define gdcmm_debug_assert(cond) ((void)0)
00094 #endif
00095
00096 // Undo warning suppression.

```

```

00115 #if defined(__clang__) && defined(__has_warning)
00116 # if __has_warning("-Wdeprecated")
00117 # pragma clang diagnostic pop
00118 # endif
00119 #endif
00120
00121 #endif

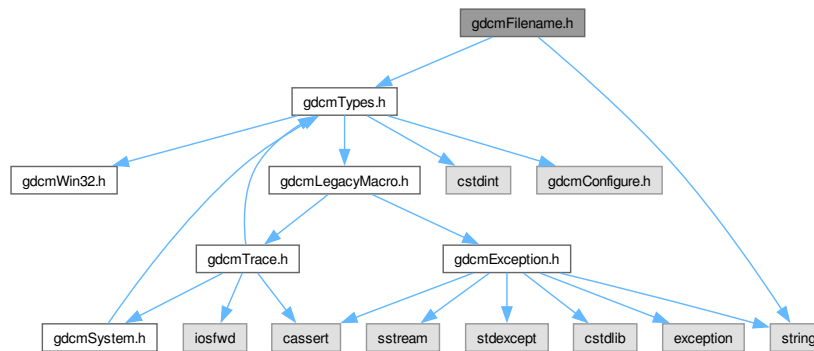
```

13.33 gdcmFilename.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmFilename.h:



Classes

- class [gdcm::Filename](#)
Class to manipulate file name's.

Namespaces

- namespace [gdcm](#)

13.34 gdcmFilename.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008

```

```

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014 #ifndef GDCMFILENAME_H
00015 #define GDCMFILENAME_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023 class GDCM_EXPORT Filename
00024 {
00025 public:
00026     Filename(const char* filename = ""):FileName(filename ? filename : ""),Path(),Conversion() {}
00027
00028     const char *GetFileName() const { return FileName.c_str(); }
00029     const char *GetPath();
00030     const char *GetName();
00031     const char *GetExtension();
00032     const char *ToUnixSlashes();
00033     const char *ToWindowsSlashes();
00034
00035     static const char *Join(const char *path, const char *filename);
00036
00037     bool IsEmpty() const { return FileName.empty(); }
00038
00039     operator const char * () const { return GetFileName(); }
00040
00041     // FIXME: I don't like this function
00042     // It hides the realpath call (maybe useful)
00043     // and it forces file to exist on the disk whereas Filename
00044     // should be independent from file existence.
00045     bool IsIdentical(Filename const &fn) const;
00046
00047     bool EndWith(const char ending[]) const;
00048
00049 private:
00050     std::string FileName;
00051     std::string Path;
00052     std::string Conversion;
00053 };
00054 } // end namespace gdcm
00055
00056 #endif //GDCMFILENAME_H

```

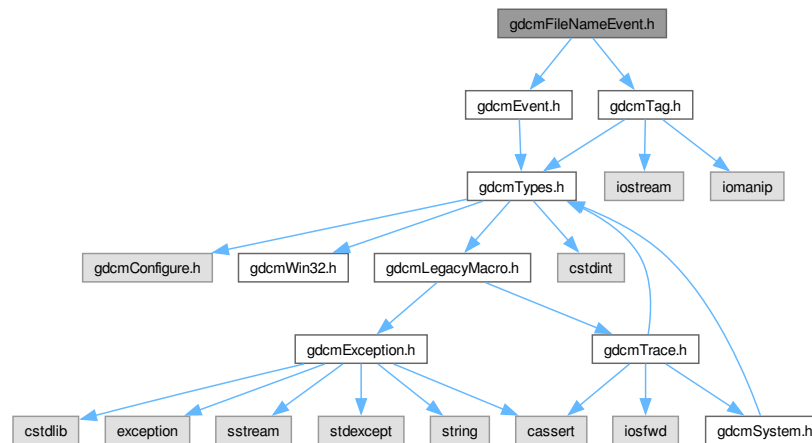
13.35 gdcmFileNameEvent.h File Reference

```

#include "gdcmEvent.h"
#include "gdcmTag.h"

```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class `gdcm::FileNameEvent`
`FileNameEvent`.

Namespaces

- namespace `gdcm`

13.36 gdcmFileNameEvent.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMFILENAMEEVENT_H
00015  #define GDCMFILENAMEEVENT_H
00016
00017  #include "gdcmEvent.h"
00018  #include "gdcmTag.h"
00019
00020  namespace gdcm
00021  {

```

```

00022
00029 class FileNameEvent : public AnyEvent
00030 {
00031 public:
00032 typedef FileNameEvent Self;
00033 typedef AnyEvent Superclass;
00034 FileNameEvent(const char *s = ""):m_FileName(s) {}
00035 ~FileNameEvent() override = default;
00036
00037 FileNameEvent(const Self&cs) : AnyEvent(s){}
00038 void operator=(const Self&) = delete;
00039
00040
00041 const char * GetEventName() const override { return "FileNameEvent"; }
00042 bool CheckEvent(const ::gdcm::Event* e) const override
00043 { return dynamic_cast<const Self*>(e) ? true : false; }
00044 ::gdcm::Event* MakeObject() const override
00045 { return new Self; }
00046
00047 void SetFileName(const char *f) { m_FileName = f; }
00048 const char *GetFileName() const { return m_FileName.c_str(); }
00049 private:
00050 std::string m_FileName;
00051 };
00052
00053
00054 } // end namespace gdcm
00055
00056 #endif //GDCMFILENAMEEVENT_H

```

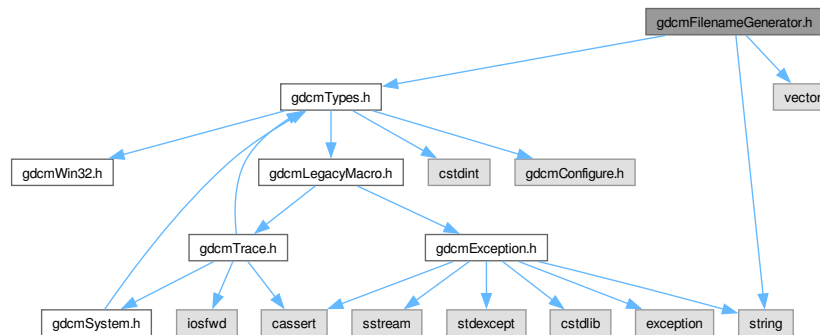
13.37 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

```
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



Classes

- class [gdcm::FilenameGenerator](#)
[FilenameGenerator](#).

Namespaces

- namespace [gdcm](#)

13.38 gdcmFilenameGenerator.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMFILENAMEGENERATOR_H
00015  #define GDCMFILENAMEGENERATOR_H
00016  00017  #include "gdcmTypes.h"
00018  #include <string>
00019  #include <vector>
00020  00021  00022  namespace gdcm
00023  {
00024  00025  00026  00027  00028  00029  00030  00031  00032  00033  00034  00035  class GDCM_EXPORT FilenameGenerator
00036  {
00037  public:
00038  FilenameGenerator():Pattern(),Prefix(),FileNames() {}
00039  ~FilenameGenerator() = default;
00040  // FIXME: already defines in gdcm::Directory
00041  typedef std::string FilenameType;
00042  typedef std::vector<FilenameType> FileNamesType;
00043  typedef FileNamesType::size_type SizeType;
00044  void SetPattern(const char *pattern) { Pattern = pattern; }
00045  const char *GetPattern() const { return Pattern.c_str(); }
00046  void SetPrefix(const char *prefix) { Prefix = prefix; }
00047  const char *GetPrefix() const { return Prefix.c_str(); }
00048  bool Generate();
00049  void SetNumberOfFileNames(SizeType nfiles);
00050  SizeType GetNumberOfFileNames() const;
00051  const char * GetFilename(SizeType n) const;
00052  FileNamesType const & GetFileNames() const { gdcm_assert( !Pattern.empty() ); return FileNames; }
00053  private:
00054  FilenameType Pattern;
00055  FilenameType Prefix;
00056  FileNamesType FileNames;
00057  };
00058  } // end namespace gdcm
00059  #endif //GDCMFILENAMEGENERATOR_H

```

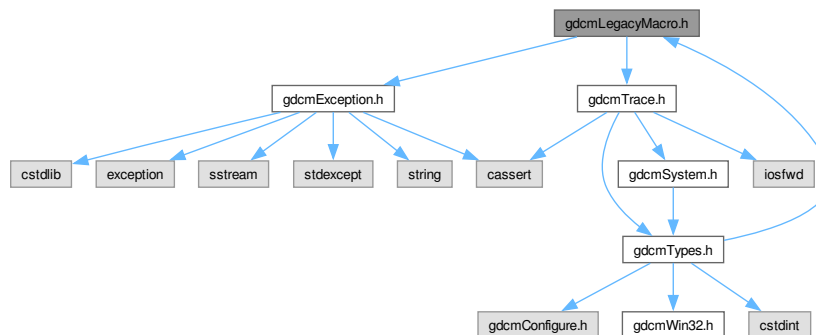
13.39 gdcmLegacyMacro.h File Reference

```

#include "gdcmException.h"
#include "gdcmTrace.h"

```

Include dependency graph for `gdcmLegacyMacro.h`:



This graph shows which files directly or indirectly include this file:



Macros

- `#define GDCM_LEGACY(method)`
- `#define GDCM_LEGACY_BODY(method, version)`
- `#define GDCM_LEGACY_REPLACED_BODY(method, version, replace)`
- `#define GDCM_NOOP_STATEMENT static_assert(true, "")`

13.39.1 Macro Definition Documentation

13.39.1.1 GDCM_LEGACY

```
#define GDCM_LEGACY(
    method)
```

Value:

`method;`

13.39.1.2 GDCM_LEGACY_BODY

```
#define GDCM_LEGACY_BODY(
    method,
    version)
```

Value:

`gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`

13.39.1.3 GDCM_LEGACY_REPLACED_BODY

```
#define GDCM_LEGACY_REPLACED_BODY(
    method,
    version,
    replace)
```

Value:

`gdcmWarningMacro`(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.)

13.39.1.4 GDCM_NOOP_STATEMENT

```
#define GDCM_NOOP_STATEMENT static_assert(true, "")
```

The `static_assert(true, "")` idiom is commonly employed for C++11 or greater to ensure that it is compile-time only check that can not be part of the binary file. This allows a macro to be used anywhere that a statement is expected, and to enforce consistent use of ; after a macro. The `static_assert` is a constexpr that can be used in places where raw statements (i.e. 'do{ } while(0)') are not allowed (i.e. after class member function definitions).

13.40 gdcmLegacyMacro.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMLEGACYMACRO_H
00015  #define GDCMLEGACYMACRO_H
00016
00017  #if !defined(GDCMTYPES_H) && !defined(SWIG)
00018  #error you need to include gdcmTypes.h instead
00019  #endif
00020
00021  #include "gdcmException.h"
00022
00023  //-----
00024  // Setup legacy code policy.
00025
00026  // Define GDCM_LEGACY macro to mark legacy methods where they are
00027  // declared in their class. Example usage:
00028  //
00029  // // @deprecated Replaced by MyOtherMethod() as of GDCM 2.0.
00030  // GDCM_LEGACY(void MyMethod());
00031  #if defined(GDCM_LEGACY_REMOVE)
00032  # define GDCM_LEGACY(method)
00033  #elif defined(GDCM_LEGACY_SILENT) || defined(SWIG)
00034  // Provide legacy methods with no warnings.
00035  # define GDCM_LEGACY(method) method;
00036  #else
00037  // Setup compile-time warnings for uses of deprecated methods if
00038  // possible on this compiler.
```

```

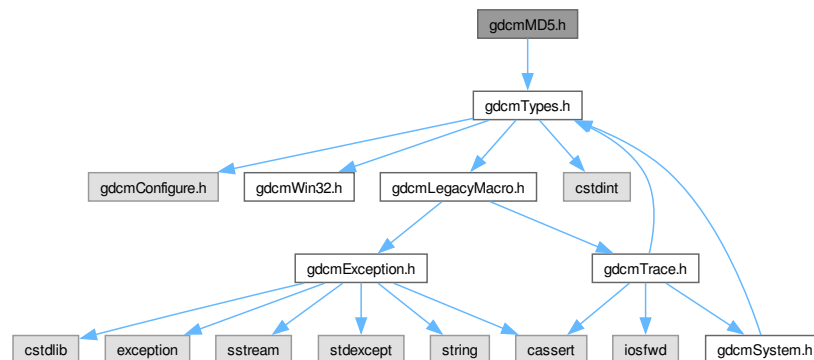
00039 # if defined(__GNUC__) && !defined(__INTEL_COMPILER) && (__GNUC__ > 3 || (__GNUC__ == 3 &&
    __GNUC_MINOR__ >= 1))
00040 # define GDCM_LEGACY(method) method __attribute__((deprecated));
00041 # elif defined(_MSC_VER) && _MSC_VER >= 1300
00042 # define GDCM_LEGACY(method) __declspec(deprecated) method;
00043 # else
00044 # define GDCM_LEGACY(method) method;
00045 # endif
00046 #endif
00047
00057 # define GDCM_NOOP_STATEMENT static_assert(true, "")
00058
00059 // Macros to create runtime deprecation warning messages in function
00060 // bodies. Example usage:
00061 //
00062 // #if !defined(GDCM_LEGACY_REMOVE)
00063 // void gdcM::MyClass::MyOldMethod()
00064 // {
00065 //     GDCM_LEGACY_BODY(gdcM::MyClass::MyOldMethod, "GDCM 2.0");
00066 // }
00067 // #endif
00068 //
00069 // #if !defined(GDCM_LEGACY_REMOVE)
00070 // void gdcM::MyClass::MyMethod()
00071 // {
00072 //     GDCM_LEGACY_REPLACED_BODY(gdcM::MyClass::MyMethod, "GDCM 2.0",
00073 //                               gdcM::MyClass::MyOtherMethod);
00074 // }
00075 // #endif
00076 #if defined(GDCM_LEGACY_REMOVE) || defined(GDCM_LEGACY_SILENT)
00077 # define GDCM_LEGACY_BODY(method, version)
00078 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace) \
00079 #else
00080 # define GDCM_LEGACY_BODY(method, version) \
00081     gdcMWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
00082 # define GDCM_LEGACY_REPLACED_BODY(method, version, replace) \
00083     gdcMWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace "
00084     instead.")
00085 #endif
00086 #include "gdcMTrace.h"
00087
00088 #endif // GDCMLEGACYMACRO_H

```

13.41 gdcMMD5.h File Reference

#include "gdcMTypes.h"

Include dependency graph for gdcMMD5.h:



Classes

- class [gdcm::MD5](#)
Class for [MD5](#).

Namespaces

- namespace [gdcm](#)

13.42 gdcmMD5.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMMD5_H
00015  #define GDCMMD5_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021  //-----
00022  class GDCM_EXPORT MD5
00023  {
00024  public :
00025  // Compute md5 from memory pointed by `pointer` of size `buf_len`
00026  static bool Compute(const char *buffer, size_t buf_len, char digest_str[33]);
00027
00028  static bool ComputeFile(const char *filename, char digest_str[33]);
00029  };
00030  } // end namespace gdcm
00031  //-----
00032  #endif //GDCMMD5_H

```

13.43 gdcmObject.h File Reference

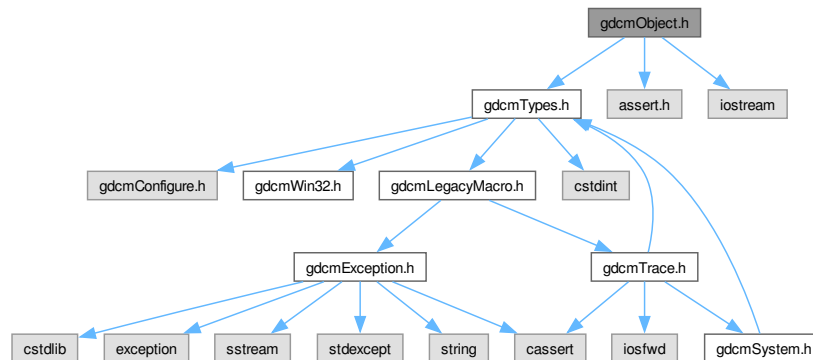
```

#include "gdcmTypes.h"
#include <assert.h>

```

```
#include <iostream>
```

Include dependency graph for `gdcmObject.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Object`
`Object`.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

13.44 gdcmObject.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  Copyright (c) 2006-2011 Mathieu Malaterre
00005  All rights reserved.

```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMOBJECT_H
00015 #define GDCMOBJECT_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <assert.h>
00020 #include <iostream> // grrrr
00021
00022 //namespace std { class ostream; }
00023 namespace gdcm
00024 {
00025
00026 template<class ObjectType> class SmartPointer;
00027
00028 class GDCM_EXPORT Object
00029 {
00030     template <class ObjectType> friend class SmartPointer;
00031     friend std::ostream& operator<<(std::ostream &os, const Object &obj);
00032
00033 public:
00034     Object():ReferenceCount(0) {}
00035
00036     // Implementation note:
00037     // If I move ~Object in the protected section I can prevent people
00038     // from writing:
00039     // SmartPointer<Object> p = new Object;
00040     // delete p; // due to SmartPointer::operator ObjectType * () const
00041     // but on the other hand one could not define an Object on the stack
00042     // Object obj;
00043     // Furthermore it would not prevent anyone from doing:
00044     // class MyObject : public Object {};
00045     // SmartPointer<MyObject> o = new MyObject;
00046     // delete o; // grrrrrr
00047     virtual ~Object() {
00048         // If your debugger reach here it means you are doing something silly
00049         // like using SmartPointer on object allocated on the stack (vs heap)
00050         gdcm_forced_assert(ReferenceCount == 0);
00051     }
00052
00053     // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.24
00054     // Set the ref count to 0
00055     // Do NOT copy the reference count !
00056     Object(const Object&):ReferenceCount(0){}
00057     void operator=(const Object&){}
00058
00059     //static Object* New() { return new Object; }
00060
00061 protected:
00062     // For the purpose of the invasive SmartPointer implementation
00063     void Register() {
00064         ReferenceCount++;
00065         gdcm_assert( ReferenceCount > 0 );
00066     }
00067     void UnRegister() {
00068         gdcm_assert( ReferenceCount > 0 );
00069         ReferenceCount--;
00070         if(!ReferenceCount)
00071         {
00072             delete this;
00073         }
00074     }
00075
00076 public:
00077     // For dealing with printing of object and polymorphism
00078     virtual void Print(std::ostream &) const {}
00079
00080 private:
00081     long ReferenceCount;
00082 };
00083
00084 //-----
00085 // function do not carry vtable. Thus define in the base class the operator
00086 // and use the member function ->Print() to call the appropriate function

```

```

00096 // NOTE: All subclass of Object needs to implement the Print function
00097 inline std::ostream& operator<<(std::ostream &os, const Object &obj)
00098 {
00099     obj.Print(os);
00100     return os;
00101 }
00102
00103 } // end namespace gdc
00104
00105 #endif //GDCMOBJECT_H

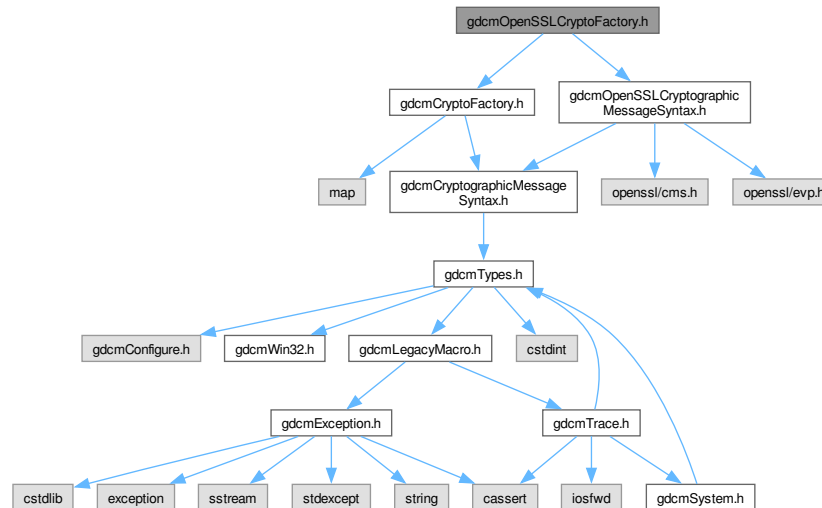
```

13.45 gdcOpenSSLCryptoFactory.h File Reference

#include "gdcCryptoFactory.h"

#include "gdcOpenSSLCryptographicMessageSyntax.h"

Include dependency graph for gdcOpenSSLCryptoFactory.h:



Classes

- class `gdc::OpenSSLCryptoFactory`

Namespaces

- namespace `gdc`

13.46 gdcmlOpenSSLCryptoFactory.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMOPENSSLCRYPTOFACTORY_H
00015 #define GDCMOPENSSLCRYPTOFACTORY_H
00016
00017 #include "gdcmlCryptoFactory.h"
00018 #include "gdcmlOpenSSLCryptographicMessageSyntax.h"
00019
00020 namespace gdcml
00021 {
00022
00023 class GDCM_EXPORT OpenSSLCryptoFactory : public CryptoFactory
00024 {
00025 public:
00026   OpenSSLCryptoFactory(CryptoLib id) : CryptoFactory(id)
00027   {
00028     gdcmlDebugMacro( "OpenSSL Factory registered." );
00029   }
00030
00031 public:
00032   CryptographicMessageSyntax* CreateCMSProvider()
00033   {
00034     InitOpenSSL();
00035     return new OpenSSLCryptographicMessageSyntax();
00036   }
00037
00038 protected:
00039   void InitOpenSSL();
00040
00041 private:
00042   OpenSSLCryptoFactory(){}
00043 };
00044
00045 } // end namespace gdcml
00046
00047 #endif //GDCMOPENSSLCRYPTOFACTORY_H

```

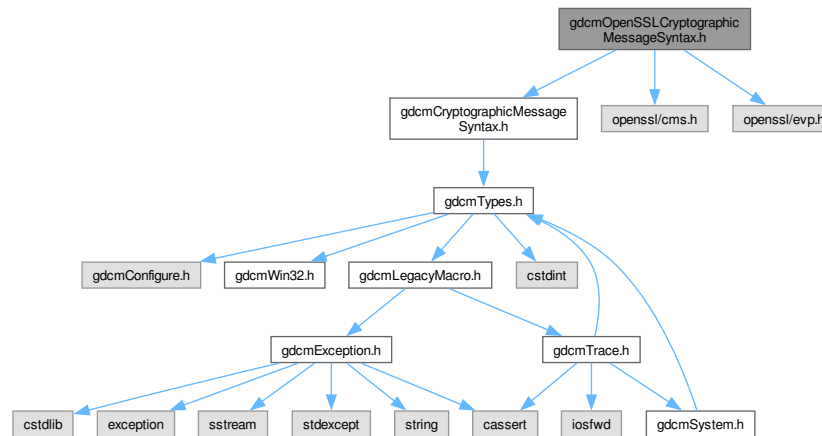
13.47 gdcmlOpenSSLCryptographicMessageSyntax.h File Reference

```

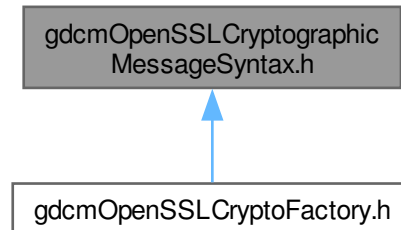
#include "gdcmlCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>

```

Include dependency graph for `gdcOpenSSLCryptographicMessageSyntax.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::OpenSSLCryptographicMessageSyntax`

Namespaces

- namespace `gdc`

13.48 gdcmlOpenSSLCryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017 #include "gdcmlCryptographicMessageSyntax.h"
00018 #include <openssl/cms.h>
00019 #include <openssl/evp.h>
00020
00021 namespace gdcml
00022 {
00023
00024 class GDCM_EXPORT OpenSSLCryptographicMessageSyntax : public CryptographicMessageSyntax
00025 {
00026 public:
00027     OpenSSLCryptographicMessageSyntax();
00028     ~OpenSSLCryptographicMessageSyntax();
00029
00030     // X.509
00031     bool ParseCertificateFile( const char *filename );
00032     bool ParseKeyFile( const char *filename );
00033
00034     // PBE
00035     bool SetPassword(const char * pass, size_t passLen);
00036
00037     void SetCipherType(CipherTypes type);
00038     CipherTypes GetCipherType() const;
00039     bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00040     bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00041
00042 private:
00043     // #ifdef GDCM_HAVE_CMS_RECIPIENT_PASSWORD
00044     // ::stack_st_X509 *recips;
00045     // #else
00046     STACK_OF(X509) *recips;
00047     // #endif
00048     ::EVP_PKEY *pkey;
00049     const EVP_CIPHER *internalCipherType;
00050     char * password;
00051     size_t passwordLength;
00052     CipherTypes cipherType;
00053
00054 private:
00055     OpenSSLCryptographicMessageSyntax(const OpenSSLCryptographicMessageSyntax&); // Not implemented.
00056     void operator=(const OpenSSLCryptographicMessageSyntax&); // Not implemented.
00057     const EVP_CIPHER *CreateCipher( CryptographicMessageSyntax::CipherTypes ciphertype);
00058
00059 };
00060
00061 } // end namespace gdcml
00062
00063 #endif //GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H

```

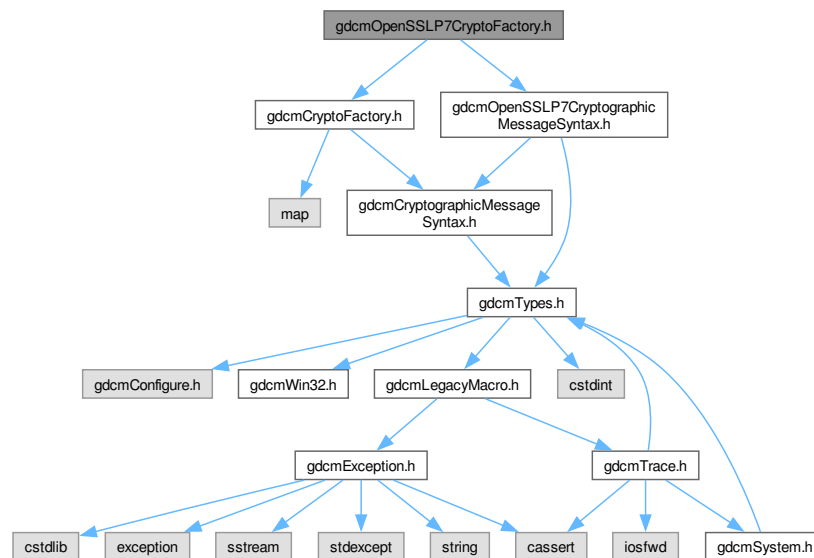
13.49 gdcmlOpenSSLP7CryptoFactory.h File Reference

```

#include "gdcmlCryptoFactory.h"
#include "gdcmlOpenSSLP7CryptographicMessageSyntax.h"

```

Include dependency graph for `gdcOpenSSL7CryptoFactory.h`:



Classes

- class `gdc::OpenSSL7CryptoFactory`

Namespaces

- namespace `gdc`

13.50 `gdcOpenSSL7CryptoFactory.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMOPENSSL7CRYPTOFACTORY_H
00015  #define GDCMOPENSSL7CRYPTOFACTORY_H
00016  00017  #include "gdcCryptoFactory.h"
00018  #include "gdcOpenSSL7CryptographicMessageSyntax.h"

```

```

00019
00020 namespace gdc
00021 {
00022 class GDCM_EXPORT OpenSSLP7CryptoFactory : public CryptoFactory
00023 {
00024 public:
00025   OpenSSLP7CryptoFactory(CryptoLib id) : CryptoFactory(id)
00026   {
00027     gdcmDebugMacro( "OpenSSL (PKCS7) Factory registered." );
00028   }
00029
00030 public:
00031   CryptographicMessageSyntax* CreateCMSProvider()
00032   {
00033     return new OpenSSLP7CryptographicMessageSyntax();
00034   }
00035
00036 private:
00037   OpenSSLP7CryptoFactory(){}
00038 };
00039 }
00040
00041 #endif //GDCMOPENSSLP7CRYPTOFACTORY_H

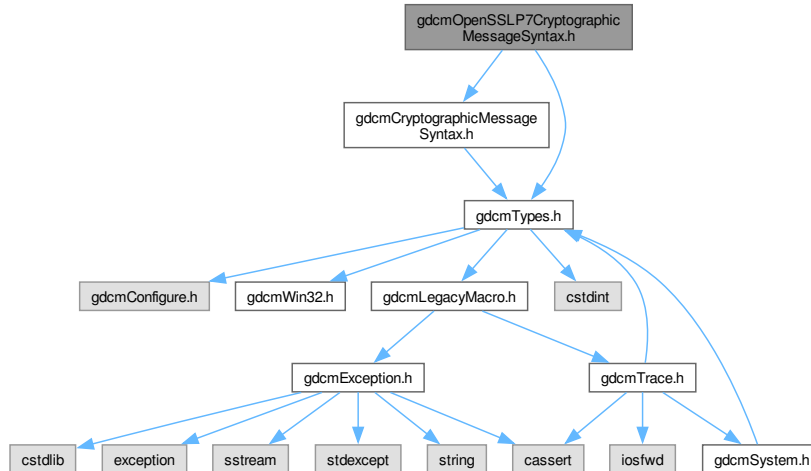
```

13.51 gdcOpenSSLP7CryptographicMessageSyntax.h File Reference

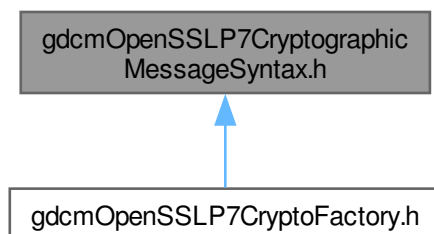
#include "gdcCryptographicMessageSyntax.h"

#include "gdcTypes.h"

Include dependency graph for gdcOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::OpenSSL7CryptographicMessageSyntax](#)

Namespaces

- namespace [gdcml](#)

13.52 gdcmlOpenSSL7CryptographicMessageSyntax.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H
00015  #define GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H
00016
00017  #include "gdcmlCryptographicMessageSyntax.h"
00018  #include "gdcmlTypes.h"
00019
00020  namespace gdcml
00021  {
00022  class CryptographicMessageSyntaxInternals;
00023  //-----
00024
00025  class GDCM_EXPORT OpenSSL7CryptographicMessageSyntax : public CryptographicMessageSyntax
00026  {
00027  public:
00028  OpenSSL7CryptographicMessageSyntax();
00029  ~OpenSSL7CryptographicMessageSyntax();
  
```

```

00039
00040 // X.509
00041 bool ParseCertificateFile( const char *filename );
00042 bool ParseKeyFile( const char *filename );
00043
00044 // PBE
00045 bool SetPassword(const char * /*pass*/, size_t /*passLen*/)
00046 {
00047     gdcmWarningMacro( "Openssl using PKCS7 does not support Password Based Encryption." );
00048     return false;
00049 }
00050
00053 void SetCipherType(CipherTypes type);
00054 CipherTypes GetCipherType() const;
00055
00057 bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00058
00060 bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;
00061
00062 private:
00063     CryptographicMessageSyntaxInternals *Internals;
00064 private:
00065     OpenSSL7CryptographicMessageSyntax(const OpenSSL7CryptographicMessageSyntax&); // Not implemented.
00066     void operator=(const OpenSSL7CryptographicMessageSyntax&); // Not implemented.
00067 };
00068 } // end namespace gdcm
00069 //-----
00070 #endif //GDCMOPENSSL7CRYPTOGRAPHICMESSAGESYNTAX_H

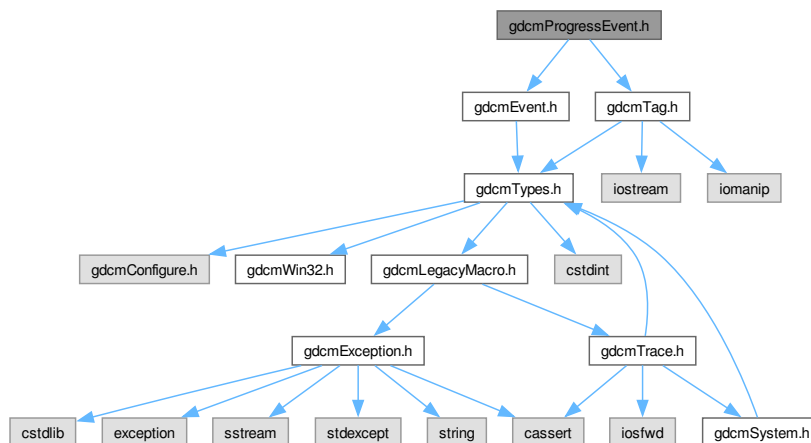
```

13.53 gdcmProcessEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for gdcmProcessEvent.h:



Classes

- class `gdcm::ProgressEvent`
 `ProgressEvent`.

Namespaces

- namespace [gdcm](#)

13.54 gdcmProgressEvent.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPROGRESSEVENT_H
00015 #define GDCMPROGRESSEVENT_H
00016
00017 #include "gdcmEvent.h"
00018 #include "gdcmTag.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class ProgressEvent : public AnyEvent
00024 {
00025 public:
00026     typedef ProgressEvent Self;
00027     typedef AnyEvent Superclass;
00028     ProgressEvent(double p = 0):m_Progress(p) {}
00029     ~ProgressEvent() override = default;
00030
00031     ProgressEvent(const Self&s) : AnyEvent(s), m_Progress(0.0) {}
00032     void operator=(const Self&) = delete;
00033
00034     const char * GetEventName() const override { return "ProgressEvent"; }
00035     bool CheckEvent(const ::gdcm::Event* e) const override
00036     { return dynamic_cast<const Self*>(e) ? true : false; }
00037     ::gdcm::Event* MakeObject() const override
00038     { return new Self; }
00039
00040     void SetProgress(double p) { m_Progress = p; }
00041     double GetProgress() const { return m_Progress; }
00042 private:
00043     double m_Progress;
00044 };
00045
00046 } // end namespace gdcm
00047
00048 #endif //GDCMPROGRESSEVENT_H

```

13.55 gdcmRegion.h File Reference

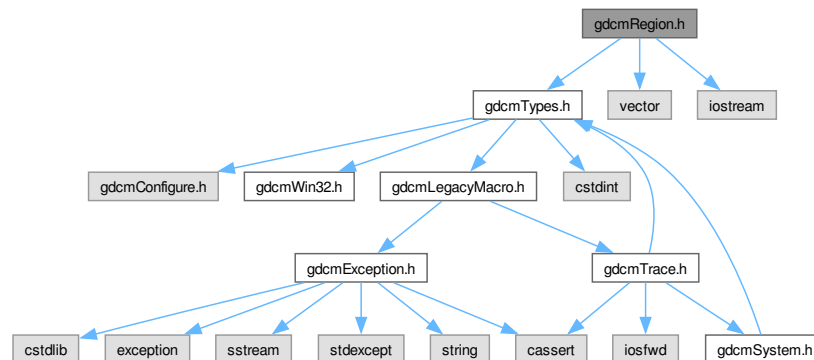
```

#include "gdcmTypes.h"
#include <vector>

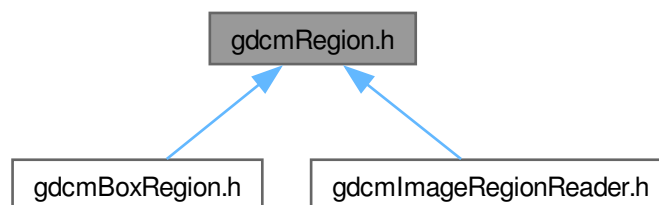
```

```
#include <iostream>
```

Include dependency graph for gdcmRegion.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Region`
Class for manipulation region.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)`

13.56 gdcmRegion.h

[Go to the documentation of this file.](#)

```

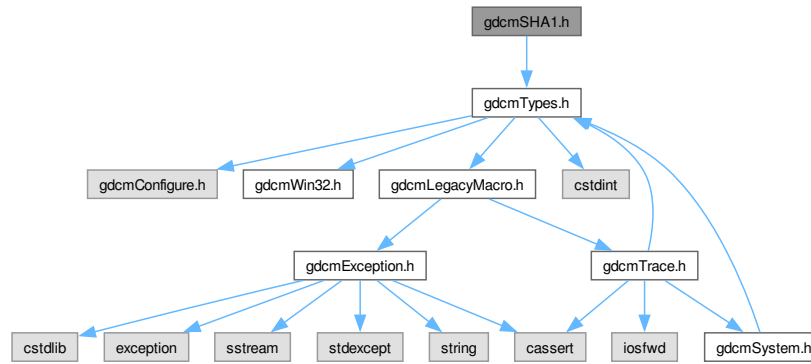
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMREGION_H
00015 #define GDCMREGION_H
00016
00017 #include "gdcmTypes.h"
00018 #include <vector>
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023 class BoxRegion;
00027 //-----
00028 class GDCM_EXPORT Region
00029 {
00030 public :
00031   Region();
00032   virtual ~Region();
00033
00034   virtual void Print(std::ostream &os = std::cout) const;
00035
00036   virtual bool Empty() const = 0;
00037
00038   virtual bool IsValid() const = 0;
00039
00040   virtual size_t Area() const = 0;
00041
00042   // implementation detail of heterogeneous container in C++
00043   virtual Region *Clone() const = 0;
00044
00045   virtual BoxRegion ComputeBoundingBox() = 0;
00046 private:
00047 };
00048 //-----
00049 inline std::ostream& operator<<(std::ostream &os, const Region&r)
00050 {
00051   r.Print( os );
00052   return os;
00053 }
00054
00055 } // end namespace gdcm
00056 //-----
00057 #endif //GDCMREGION_H

```


13.57 gdcmSHA1.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmSHA1.h:



Classes

- class [gdcm::SHA1](#)
Class for [SHA1](#).

Namespaces

- namespace [gdcm](#)

13.58 gdcmSHA1.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSHA1_H
00015  #define GDCMSHA1_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {

```

```

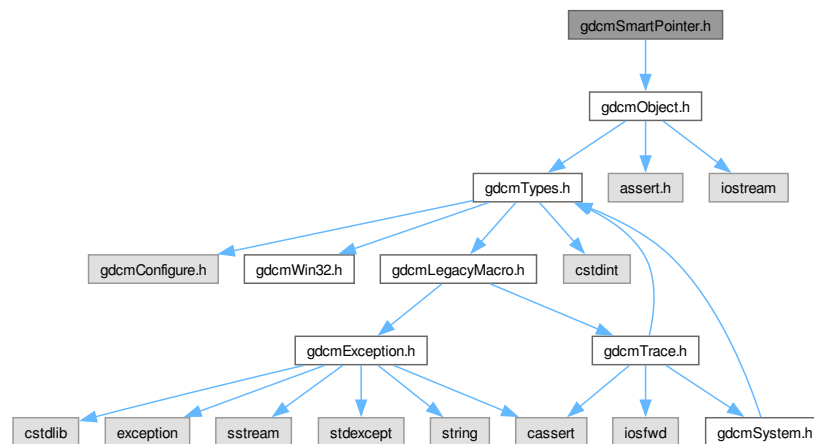
00021 //-----
00022 class SHA1Internals;
00032 class GDCM_EXPORT SHA1
00033 {
00034 public :
00035     SHA1();
00036     ~SHA1();
00037     SHA1(const SHA1&) = delete;
00038     void operator=(const SHA1&) = delete;
00039
00040     static bool Compute(const char *buffer, unsigned long buf_len, char digest_str[20*2+1]);
00041
00042     static bool ComputeFile(const char *filename, char digest_str[20*2+1]);
00043
00044 private:
00045     SHA1Internals *Internals;
00046 };
00047 } // end namespace gdcmm
00048 //-----
00049 #endif //GDCMSHA1_H

```

13.59 gdcmmSmartPointer.h File Reference

#include "gdcmmObject.h"

Include dependency graph for gdcmmSmartPointer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::SmartPointer< ObjectType >`
Class for Smart Pointer.

Namespaces

- namespace `gdcm`

13.60 gdcmSmartPointer.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSMARTPOINTER_H
00015 #define GDCMSMARTPOINTER_H
00016
00017 #include "gdcmObject.h"
00018
00019 namespace gdcm
00020 {
00021     template<class ObjectType>
00022     class SmartPointer
00023     {
00024     public:
00025         SmartPointer():Pointer(nullptr) {}
00026         SmartPointer(const SmartPointer<ObjectType>& p):Pointer(p.Pointer)
00027         { Register(); }
00028         SmartPointer(ObjectType* p):Pointer(p)
00029         { Register(); }
00030         SmartPointer(ObjectType const & p)
00031         {
00032             Pointer = const_cast<ObjectType*>(&p);
00033             Register();
00034         }
00035         ~SmartPointer() {
00036             UnRegister();
00037             Pointer = nullptr;
00038         }
00039
00040         ObjectType *operator -> () const
00041         { return Pointer; }
00042
00043         ObjectType& operator * () const
00044         {
00045             gdcm_assert( Pointer );
00046             return *Pointer;
00047         }
00048
00049         operator ObjectType * () const
00050         { return Pointer; }
00051
00052         SmartPointer &operator = (SmartPointer const &r)
00053         { return operator = (r.Pointer); }
00054
00055         SmartPointer &operator = (ObjectType *r)
00056         {
00057             // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.22
00058             // DO NOT CHANGE THE ORDER OF THESE STATEMENTS!
00059             // (This order properly handles self-assignment)
00060             // (This order also properly handles recursion, e.g., if a ObjectType contains SmartPointer<ObjectType>s)
00061             if( Pointer != r )
00062             {
00063                 ObjectType* old = Pointer;
00064                 Pointer = r;
00065                 Register();
00066             }
00067         }
00068     };
00069
00070     SmartPointer &operator = (SmartPointer const &r)
00071     { return operator = (r.Pointer); }
00072
00073     SmartPointer &operator = (ObjectType *r)
00074     {
00075         // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.22
00076         // DO NOT CHANGE THE ORDER OF THESE STATEMENTS!
00077         // (This order properly handles self-assignment)
00078         // (This order also properly handles recursion, e.g., if a ObjectType contains SmartPointer<ObjectType>s)
00079         if( Pointer != r )
00080         {
00081             ObjectType* old = Pointer;
00082             Pointer = r;
00083             Register();
00084         }
00085     }
00086 }

```

```

00087     if ( old ) { old->UnRegister(); }
00088     }
00089     return *this;
00090 }
00091
00092 SmartPointer &operator = (ObjectType const &r)
00093 {
00094     ObjectType* tmp = const_cast<ObjectType*>(&r);
00095     return operator = (tmp);
00096 }
00097
00098 ObjectType *GetPointer() const
00099 { return Pointer; }
00100
00101 private:
00102 void Register()
00103 {
00104     if(Pointer) Pointer->Register();
00105 }
00106
00107 void UnRegister()
00108 {
00109     if(Pointer) Pointer->UnRegister();
00110 }
00111
00112 ObjectType* Pointer;
00113 };
00114
00115 // end namespace gdcm
00116 #endif //GDCMSMARTPOINTER_H

```

13.61 gdcmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [gdcm::static_assert_test< x >](#)
- struct [gdcm::STATIC_ASSERTION_FAILURE< true >](#)

Namespaces

- namespace [gdcm](#)

Macros

- #define [GDCM_DO_JOIN\(X, Y\)](#)
- #define [GDCM_DO_JOIN2\(X, Y\)](#)
- #define [GDCM_JOIN\(X, Y\)](#)
- #define [GDCM_STATIC_ASSERT\(B\)](#)

The GDCM_JOIN + LINE is needed to create a uniq identifier.

13.61.1 Macro Definition Documentation

13.61.1.1 GDCM_DO_JOIN

```
#define GDCM_DO_JOIN(  
    X,  
    Y)
```

Value:

`GDCM_DO_JOIN2(X,Y)`

13.61.1.2 GDCM_DO_JOIN2

```
#define GDCM_DO_JOIN2(  
    X,  
    Y)
```

Value:

`X##Y`

13.61.1.3 GDCM_JOIN

```
#define GDCM_JOIN(  
    X,  
    Y)
```

Value:

`GDCM_DO_JOIN(X, Y)`

13.61.1.4 GDCM_STATIC_ASSERT

```
#define GDCM_STATIC_ASSERT(  
    B)
```

Value:

```
typedef ::gdcm::static_assert_test<\n    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\n    GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + LINE is needed to create a uniq identifier.

13.62 gdcmStaticAssert.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSTATICASSERT_H
00015 #define GDCMSTATICASSERT_H
00016
00017
00018 // the following was shamelessly borrowed from BOOST static assert:
00019 namespace gdcm
00020 {
00021     template <bool x>
00022     struct STATIC_ASSERTION_FAILURE;
00023
00024     template <>
00025     struct STATIC_ASSERTION_FAILURE<true> { enum { value = 1 }; };
00026
00027     template <int x>
00028     struct static_assert_test {};
00029 }
00030
00031 #define GDCM_JOIN( X, Y ) GDCM_DO_JOIN( X, Y )
00032 #define GDCM_DO_JOIN( X, Y ) GDCM_DO_JOIN2(X,Y)
00033 #define GDCM_DO_JOIN2( X, Y ) X##Y
00034
00035 #define GDCM_STATIC_ASSERT( B ) \
00036     typedef ::gdcm::static_assert_test<\
00037         sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >>>\
00038             GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
00039
00040
00041
00042 /* Example of use:
00043 *
00044 * template <class T>
00045 * struct must_not_be_instantiated
00046 * {
00047 *     // this will be triggered if this type is instantiated
00048 *     GDCM_STATIC_ASSERT(sizeof(T) == 0);
00049 * };
00050 *
00051 */
00052 #endif // GDCMSTATICASSERT_H

```

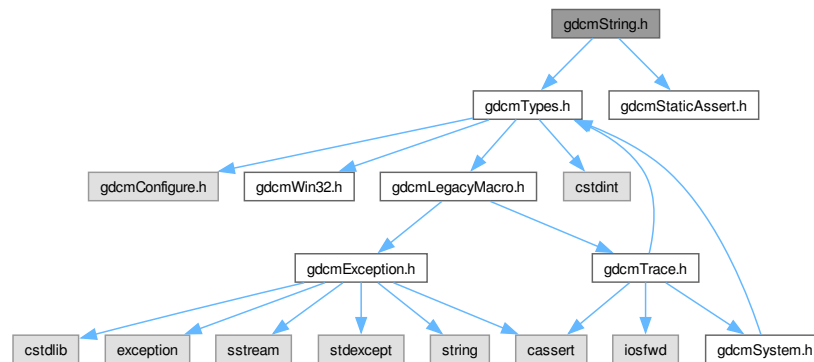
13.63 gdcmString.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"

```

Include dependency graph for gdcmString.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::String< TDelimiter, TMaxLength, TPadChar > String](#).

Namespaces

- namespace [gdcm](#)

Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & [gdcm::operator>>](#) (std::istream &is, [String< TDelimiter, TMaxLength, TPadChar >](#) &ms)

13.64 gdcmString.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSTRING_H
00015 #define GDCMSTRING_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmStaticAssert.h"
00019
00020 namespace gdcm
00021 {
00022
00023 template <char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
00024 class /*GDCM_EXPORT*/ String : public std::string /* PLEASE do not export me */
00025 {
00026 // UI wants \0 for pad character, while ASCII ones wants space char... do not allow anything else
00027 GDCM_STATIC_ASSERT( TPadChar == ' ' || TPadChar == 0 );
00028
00029 public:
00030 // typedef are not inherited:
00031 typedef std::string::value_type value_type;
00032 typedef std::string::pointer pointer;
00033 typedef std::string::reference reference;
00034 typedef std::string::const_reference const_reference;
00035 typedef std::string::size_type size_type;
00036 typedef std::string::difference_type difference_type;
00037 typedef std::string::iterator iterator;
00038 typedef std::string::const_iterator const_iterator;
00039 typedef std::string::reverse_iterator reverse_iterator;
00040 typedef std::string::const_reverse_iterator const_reverse_iterator;
00041
00042 String(): std::string() {}
00043 String(const value_type* s): std::string(s)
00044 {
00045 if( size() % 2 )
00046 {
00047 push_back( TPadChar );
00048 }
00049 }
00050 String(const value_type* s, size_type n): std::string(s, n)
00051 {
00052 // We are being passed a const char* pointer, so s[n] == 0 (guaranteed!)
00053 if( n % 2 )
00054 {
00055 push_back( TPadChar );
00056 }
00057 }
00058 String(const std::string& s, size_type pos=0, size_type n=npos):
00059 std::string(s, pos, n)
00060 {
00061 // FIXME: some users might already have padded the string 's' with a trailing \0...
00062 if( size() % 2 )
00063 {
00064 push_back( TPadChar );
00065 }
00066 }
00067
00068 operator const char*() const { return this->c_str(); }
00069
00070 bool IsValid() const {
00071 // Check Length:
00072 size_type l = size();
00073 if( l > TMaxLength ) return false;

```



```

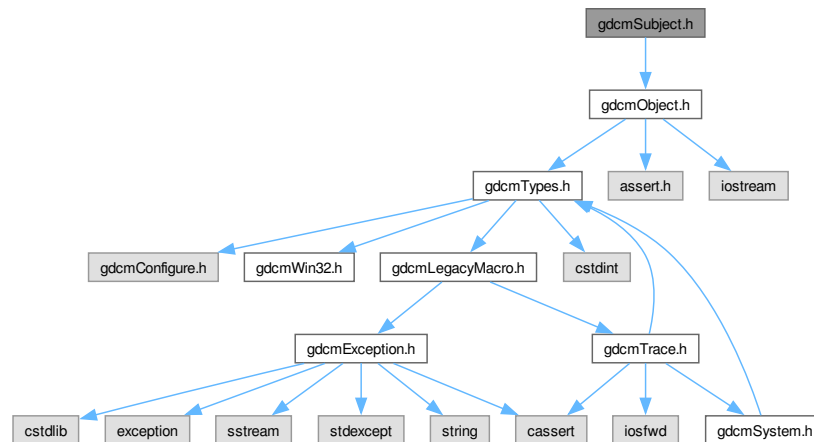
00084     return true;
00085 }
00086
00087 gdcm::String<TDelimiter, TMaxLength, TPadChar> Truncate() const {
00088     if( IsValid() ) return *this;
00089     std::string str = *this; // copy
00090     str.resize( TMaxLength );
00091     return str;
00092 }
00093
00096 std::string Trim() const {
00097     std::string str = *this; // copy
00098     std::string::size_type pos1 = str.find_first_not_of(' ');
00099     std::string::size_type pos2 = str.find_last_not_of(' ');
00100     str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
00101         (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
00102     return str;
00103 }
00104
00105 static std::string Trim(const char *input) {
00106     if( !input ) return "";
00107     std::string str = input;
00108     std::string::size_type pos1 = str.find_first_not_of(' ');
00109     std::string::size_type pos2 = str.find_last_not_of(' ');
00110     str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
00111         (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
00112     return str;
00113 }
00114 };
00115 template <char TDelimiter, unsigned int TMaxLength, char TPadChar>
00116 inline std::istream& operator>(std::istream &is, String<TDelimiter,TMaxLength,TPadChar> &ms)
00117 {
00118     if(is)
00119     {
00120         std::getline(is, ms, TDelimiter);
00121         // no such thing as std::get where the delim char would be left, so I need to manually add it back...
00122         // hopefully this is the right thing to do (no overhead)
00123         if( !is.eof() ) is.putback( TDelimiter );
00124     }
00125     return is;
00126 }
00127 //template <char TDelimiter = EOF, unsigned int TMaxLength = 64, char TPadChar = ' '>
00128 //String String::Trim() const
00129 //{
00130 //    String s;
00131 //    return s;
00132 //}
00133
00134 } // end namespace gdcm
00135
00136 #endif //GDCMSTRING_H

```

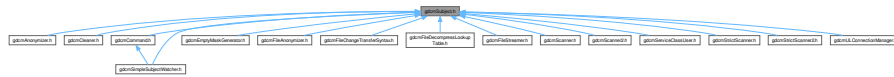
13.65 gdcmSubject.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for gdcmSubject.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Subject](#)
[Subject](#).

Namespaces

- namespace [gdcm](#)

13.66 gdcmSubject.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004

```

```

00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMSUBJECT_H
00015 #define GDCMSUBJECT_H
00016
00017 #include "gdcmObject.h"
00018
00019 namespace gdcm
00020 {
00021 class Event;
00022 class Command;
00023 class SubjectInternals;
00028 class GDCM_EXPORT Subject : public Object
00029 {
00030 public:
00031 Subject();
00032 ~Subject() override;
00033
00042 unsigned long AddObserver(const Event & event, Command *);
00043 unsigned long AddObserver(const Event & event, Command *) const;
00044
00050 Command* GetCommand(unsigned long tag);
00051
00053 void InvokeEvent( const Event & );
00054
00057 void InvokeEvent( const Event & ) const;
00058
00060 void RemoveObserver(unsigned long tag);
00061
00063 void RemoveAllObservers();
00064
00066 bool HasObserver( const Event & event ) const;
00067
00068 protected:
00069
00070 private:
00071 SubjectInternals *Internals;
00072 private:
00073 };
00074
00075 } // end namespace gdcm
00076
00077 #endif //GDCMSUBJECT_H

```

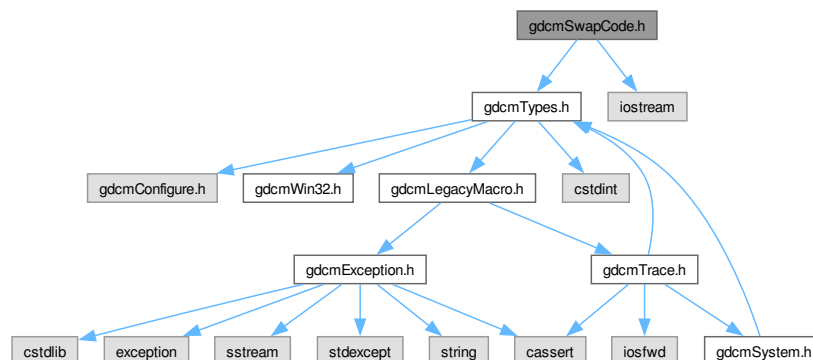
13.67 gdcmSwapCode.h File Reference

```

#include "gdcmTypes.h"
#include <iostream>

```

Include dependency graph for `gdcmSwapCode.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::SwapCode`
`SwapCode` representation.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

13.68 gdcmSwapCode.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.

```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMSWAPCODE_H
00015 #define GDCMSWAPCODE_H
00016
00017 #include "gdcmTypes.h"
00018 #include <iostream>
00019
00020 namespace gdcm
00021 {
00022
00026 class GDCM_EXPORT SwapCode
00027 {
00028 public:
00029 typedef enum {
00030     Unknown = 0,
00031     LittleEndian = 1234,
00032     BigEndian = 4321,
00033     BadLittleEndian = 3412,
00034     BadBigEndian = 2143
00035 } SwapCodeType;
00036
00037 operator SwapCodeType() const { return SwapCodeValue; }
00038 SwapCode(SwapCodeType sc = Unknown):SwapCodeValue(sc) { }
00039 static const char* GetSwapCodeString(SwapCode const & sc);
00040
00041 friend std::ostream& operator<<(std::ostream& os, const SwapCode& sc);
00042 protected:
00043 static int GetIndex(SwapCode const & sc);
00044
00045 private:
00046 SwapCodeType SwapCodeValue;
00047 };
00048 //-----
00049 inline std::ostream& operator<<(std::ostream& os, const SwapCode& sc)
00050 {
00051     os << SwapCode::GetSwapCodeString(sc);
00052     return os;
00053 }
00054
00055 } // end namespace gdcm
00056
00057 #endif //GDCMSWAPCODE_H

```

13.69 gdcmSwapper.h File Reference

```

#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"

```



```

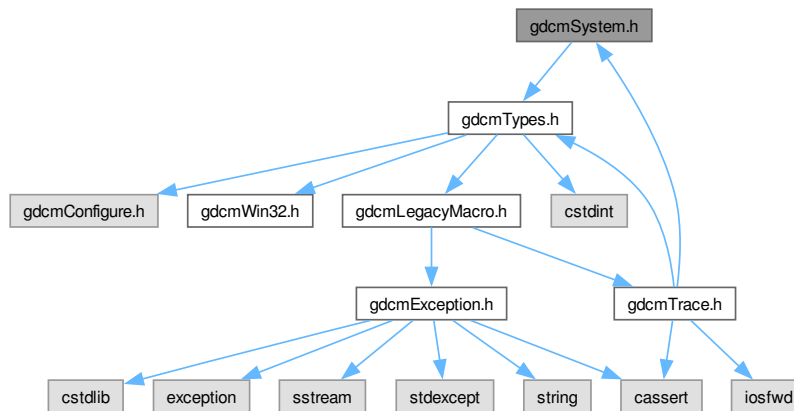
00012
00013 =====*/
00014 #ifndef GDCMSWAPPER_H
00015 #define GDCMSWAPPER_H
00016
00017 #include "gdcmSwapCode.h"
00018
00019 namespace gdcm
00020 {
00021
00022
00023 #ifdef GDCM_WORDS_BIGENDIAN
00024 class SwapperDoOp
00025 {
00026 public:
00027     template <typename T> static T Swap(T val) {return val;}
00028     template <typename T> static void SwapArray(T *, size_t ) {}
00029 };
00030
00031 class SwapperNoOp
00032 {
00033 public:
00034     template <typename T> static T Swap(T val);
00035     template <typename T>
00036     static void SwapArray(T *array, size_t n)
00037     {
00038         // TODO: need to unroll loop:
00039         for(size_t i = 0; i < n; ++i)
00040         {
00041             array[i] = Swap<T>(array[i]);
00042         }
00043     };
00044 };
00045 #else
00046 class SwapperNoOp
00047 {
00048 public:
00049     template <typename T> static T Swap(T val) {return val;}
00050     template <typename T> static void SwapArray(T *, size_t ) {}
00051 };
00052
00053 class SwapperDoOp
00054 {
00055 public:
00056     template <typename T> static T Swap(T val);
00057     template <typename T>
00058     static void SwapArray(T *array, size_t n)
00059     {
00060         // TODO: need to unroll loop:
00061         for(size_t i = 0; i < n; ++i)
00062         {
00063             array[i] = Swap<T>(array[i]);
00064         }
00065     };
00066 };
00067 #endif
00068
00069
00070 } // end namespace gdcm
00071
00072 #include "gdcmSwapper.txx"
00073
00074 #endif //GDCMSWAPPER_H

```

13.71 gdcmSystem.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSystem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::System](#)
Class to do system operation.

Namespaces

- namespace [gdcm](#)

13.72 gdcmSystem.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
  
```



```

00008
00009     This software is distributed WITHOUT ANY WARRANTY; without even
00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00015 #ifndef GDCMSYSTEM_H
00016 #define GDCMSYSTEM_H
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class GDCM_EXPORT System
00023 {
00024 public:
00025     static bool MakeDirectory(const char *path);
00026     static bool FileExists(const char* filename);
00027     static bool FileIsDirectory(const char* name);
00028     static bool FileIsSymlink(const char* name);
00029     static bool RemoveFile(const char* source);
00030     static bool DeleteDirectory(const char *source);
00031
00032     static std::wstring ConvertToUNC(const char *utf8path);
00033
00034     static const char *GetLastSystemError();
00035
00036     static size_t FileSize(const char* filename);
00037
00038     static time_t FileTime(const char* filename);
00039
00040     static const char *GetCurrentProcessFileName();
00041
00042     static const char *GetCurrentModuleFileName();
00043
00044     static const char *GetCurrentResourcesDirectory();
00045
00046     // TODO some system calls
00047     // Chdir
00048     // copy a file
00049
00050     static bool GetHostName(char hostname[255]);
00051
00052     // In the following the size '22' is explicitly listed. You need to pass in
00053     // at least 22bytes of array. If the string is an output it will be
00054     // automatically padded ( array[21] == 0 ) for you.
00055     // Those functions: GetCurrentDateTime / FormatDateTime / ParseDateTime do
00056     // not return the &YYZZ part of the DT structure as defined in DICOM PS 3.5 -
00057     // 2008 In this case it is simple to split the date[22] into a DA and TM
00058     // structure
00059
00060     static bool GetCurrentDateTime(char date[22]);
00061
00062     static bool FormatDateTime(char date[22], time_t t, long milliseconds = 0);
00063
00064     static bool ParseDateTime(time_t &timep, const char date[22]);
00065
00066     static bool ParseDateTime(time_t &timep, long &milliseconds, const char date[22]);
00067
00068     static const char *GetTimezoneOffsetFromUTC();
00069
00070     static size_t EncodeBytes(char *out, const unsigned char *data, int size);
00071
00072     static int StrCaseCmp(const char *s1, const char *s2);
00073     static int StrNCaseCmp(const char *s1, const char *s2, size_t n);
00074
00075     static const char * GetCWD();
00076
00077     static char *StrTokR(char *ptr, const char *sep, char **end);
00078
00079     static char *StrSep(char **stringp, const char *delim);
00080
00081     static const char *GetLocaleCharset();
00082
00083     /*
00084     static void SetArgv0(const char *);
00085     static const char* GetArgv0();
00086     */
00087
00088

```

```

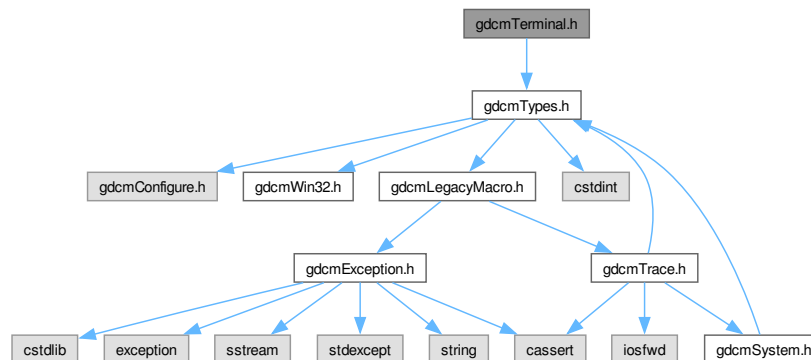
00142 protected:
00143     static bool GetPermissions(const char* file, unsigned short& mode);
00144     static bool SetPermissions(const char* file, unsigned short mode);
00145
00146 private:
00147 };
00148
00149 } // end namespace gdcm
00150
00151 #endif //GDCMSYSTEM_H

```

13.73 gdcmTerminal.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmTerminal.h:



Namespaces

- namespace `gdcm`
- namespace `gdcm::terminal`
Class for Terminal.

Enumerations

- enum `gdcm::terminal::Attribute` {
`gdcm::terminal::reset` = 0 ,
`gdcm::terminal::bright` = 1 ,
`gdcm::terminal::dim` = 2 ,
`gdcm::terminal::underline` = 3 ,
`gdcm::terminal::blink` = 5 ,
`gdcm::terminal::reverse` = 7 ,
`gdcm::terminal::hidden` = 8 }

- `enum gdcm::terminal::Color {
gdcm::terminal::black = 0 ,
gdcm::terminal::red ,
gdcm::terminal::green ,
gdcm::terminal::yellow ,
gdcm::terminal::blue ,
gdcm::terminal::magenta ,
gdcm::terminal::cyan ,
gdcm::terminal::white }`
- `enum gdcm::terminal::Mode {
gdcm::terminal::CONSOLE = 0 ,
gdcm::terminal::VT100 }`

Functions

- `GDCM_EXPORT std::string gdcm::terminal::setAttribute (Attribute att)`
- `GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color c)`
- `GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color c)`
- `GDCM_EXPORT void gdcm::terminal::setmode (Mode m)`

13.74 gdcmTerminal.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMTERMINAL_H
00015  #define GDCMTERMINAL_H
00016
00017  #include "gdcmTypes.h"
00018
00019
00020  namespace gdcm
00021  {
00022  //-----
00023
00024  namespace terminal
00025  {
00026  {
00027  typedef enum
00028  {
00029  CONSOLE = 0,
00030  VT100
00031  } Mode;
00032  typedef enum
00033  {
00034  black = 0,
00035  red,
00036  green,
00037  yellow, // brown ??
00038  blue,
00039  magenta,
00040  cyan,

```

```

00046     white
00047     } Color;
00048     typedef enum
00049     {
00050         reset    = 0,
00051         bright   = 1, // bold
00052         dim      = 2,
00053         underline = 3,
00054         blink    = 5,
00055         reverse  = 7,
00056         hidden   = 8
00057     } Attribute;
00058     GDCM_EXPORT std::string setattribute( Attribute att );
00059     GDCM_EXPORT std::string setfgcolor( Color c );
00060     GDCM_EXPORT std::string setbgcolor( Color c );
00061     GDCM_EXPORT void setmode( Mode m);
00062 }
00063
00064 } // end namespace gdcms
00065 //-----
00066 #endif //GDCMTERMINAL_H

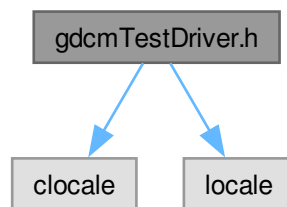
```

13.75 gdcmsTestDriver.h File Reference

```
#include <locale>
```

```
#include <locale>
```

Include dependency graph for gdcmsTestDriver.h:



13.76 gdcmsTestDriver.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/

```

```

00014 // This header is included by all the C++ test drivers in GDCM.
00015 #ifndef GDCMTESTDRIVER_H
00016 #define GDCMTESTDRIVER_H
00017
00018 // CREATE_TEST_SOURCELIST supports the flag EXTRA_INCLUDE but only one per call.
00019 // So there is no way to specify we want to include two files... instead
00020 // gather the #include in a single file and include that one...
00021 #include <locale> // C setlocale()
00022 #include <locale> // C++ locale
00023
00024 #endif // GDCMTESTDRIVER_H

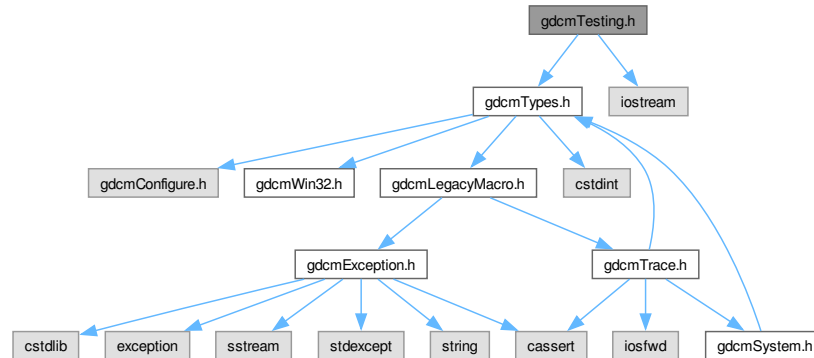
```

13.77 gdcmTesting.h File Reference

#include "gdcmTypes.h"

#include <iostream>

Include dependency graph for gdcmTesting.h:



Classes

- class [gdcm::Testing](#)
class for testing

Namespaces

- namespace [gdcm](#)

13.78 gdcmTesting.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004

```

```

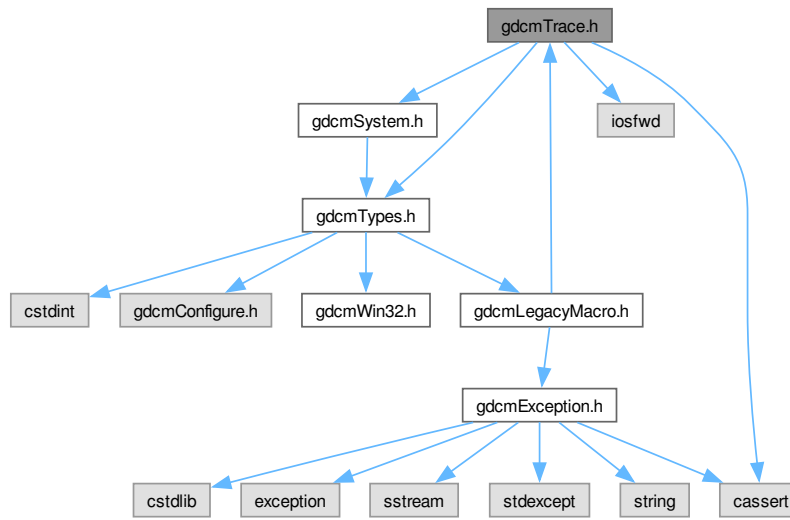
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMTESTING_H
00015 #define GDCMTESTING_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023 //-----
00031 class GDCM_EXPORT Testing
00032 {
00033 public :
00034     Testing() = default;
00035     ~Testing() = default;
00036
00042     static bool ComputeMD5(const char *buffer, size_t buf_len,
00043         char digest_str[33]);
00044     static bool ComputeFileMD5(const char *filename, char digest_str[33]);
00045
00047     void Print(std::ostream &os = std::cout);
00048
00050     static const char * const * GetFileNames();
00051     static unsigned int GetNumberOfFileNames();
00052     static const char * GetFileName(unsigned int file);
00053
00055     typedef const char* const (*MediaStorageDataFilesType)[2];
00056     static MediaStorageDataFilesType GetMediaStorageDataFiles();
00057     static unsigned int GetNumberOfMediaStorageDataFiles();
00058     static const char * const * GetMediaStorageDataFile(unsigned int file);
00059     static const char * GetMediaStorageFromFile(const char *filepath);
00060
00063     typedef const char* const (*MD5DataImagesType)[2];
00064     static MD5DataImagesType GetMD5DataImages();
00065     static unsigned int GetNumberOfMD5DataImages();
00066     static const char * const * GetMD5DataImage(unsigned int file);
00067     static const char * GetMD5FromFile(const char *filepath);
00068
00071     static const char * GetMD5FromBrokenFile(const char *filepath);
00072
00075     static std::streamoff GetStreamOffsetFromFile(const char *filepath);
00076
00080     static std::streamoff GetSelectedTagsOffsetFromFile(const char *filepath);
00081
00085     static std::streamoff GetSelectedPrivateGroupOffsetFromFile(const char *filepath);
00086
00091     static int GetLossyFlagFromFile(const char *filepath);
00092
00094     static const char * GetDataRoot();
00095
00097     static const char * GetDataExtraRoot();
00098
00100     static const char * GetPixelSpacingDataRoot();
00101
00104     static const char * GetTempDirectory(const char * subdir = nullptr);
00105
00107     static const wchar_t *GetTempDirectoryW(const wchar_t * subdir = nullptr);
00108
00110     static const char * GetTempFilename(const char *filename, const char * subdir = nullptr);
00111
00113     static const wchar_t* GetTempFilenameW(const wchar_t *filename, const wchar_t* subdir = nullptr);
00114
00115     static const char *GetSourceDirectory();
00116 };
00117 } // end namespace gdcm
00118 //-----
00119 #endif //GDCMTESTING_H

```

13.79 gdcmTrace.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSystem.h"
#include <iosfwd>
#include <cassert>
```

Include dependency graph for gdcmTrace.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Trace`
`Trace`.

Namespaces

- namespace `gdcm`

Macros

- `#define GDCM_FUNCTION "<unknown>"`
- `#define gdcmAssertAlwaysMacro(arg)`
AssertAlways.
- `#define gdcmAssertMacro(arg)`
Assert.
- `#define gdcmDebugMacro(msg)`
Debug.
- `#define gdcmErrorMacro(msg)`
Error this is pretty bad, more than just warning It could mean lost of data, something not handle...
- `#define gdcmWarningMacro(msg)`
Warning.

13.79.1 Macro Definition Documentation

13.79.1.1 GDCM_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

13.79.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(  
    arg)
```

Value:

`gdcmAssertMacro(arg)`

AssertAlways.

Parameters

arg	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro("my message" && 2 < 3)</code>
-----	---

Referenced by `gdcm::DataElement::GetValue()`, `gdcm::DataElement::GetValue()`, `gdcm::BasicOffsetTable::Read()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::DataSet::Replace()`, `gdcm::DataSet::ReplaceEmpty()`, and `gdcm::VR::Write()`.

13.79.1.3 gdcmAssertMacro

```
#define gdcmAssertMacro(
    arg)
```

Value:

```
{
    if( !(arg) )
    {
        std::ostringstream osmacro;
        osmacro << "Assert: In " << FILE__ ", line " << LINE__ \
        << ", function " << GDCM_FUNCTION \
        << "\n\n";
        std::ostream &_os = gdcm::Trace::GetErrorStream();
        _os << osmacro.str() << std::endl;
        assert ( arg );
    }
}
GDCM_NOOP_STATEMENT
```

Assert.

Parameters

arg	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro("my message" && 2 < 3)</code>
-----	---

Referenced by [gdcm::PixelFormat::SetSamplesPerPixel\(\)](#).

13.79.1.4 gdcmDebugMacro

```
#define gdcmDebugMacro(
    msg)
```

Value:

```
{
    if( gdcm::Trace::GetDebugFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Debug: In " << FILE__ ", line " << LINE__ \
        << ", function " << GDCM_FUNCTION << '\n' \
        << "Last system error was: " \
        << gdcm::System::GetLastSystemError() << '\n' << msg; \
        std::ostream &_os = gdcm::Trace::GetDebugStream();
        _os << osmacro.str() << "\n\n" << std::endl;
    }
}
GDCM_NOOP_STATEMENT
```

Debug.

Parameters

msg	message part
-----	--------------

Referenced by [gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory\(\)](#), [gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory\(\)](#), [gdcm::Item::Read\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), [gdcm::VR::Read\(\)](#), [gdcm::SequenceOfFragments::ReadPreValue\(\)](#), and [gdcm::SequenceOfFragments::ReadValue\(\)](#).

13.79.1.5 `gdcmErrorMacro`

```
#define gdcmErrorMacro(
    msg)
```

Value:

```
{
    if( gdcm::Trace::GetErrorFlag\(\) )
    {
        std::ostringstream osmacro;
        osmacro << "Error: In " << \_\_FILE\_\_ ", line " << \_\_LINE\_\_
            << ", function " << GDCM\_FUNCTION << "\n";
        << msg << "\n\n";
        std::ostream &_os = gdcm::Trace::GetErrorStream\(\);
        _os << osmacro.str() << std::endl;
    }
}
GDCM\_NOOP\_STATEMENT
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

Parameters

msg	second message part
-----	---------------------

Referenced by [gdcm::CommandDataSet::Insert\(\)](#), [gdcm::DataSet::Insert\(\)](#), [gdcm::FileMetaInformation::Insert\(\)](#), [gdcm::Item::Read\(\)](#), and [gdcm::Fragment::ReadBacktrack\(\)](#).

13.79.1.6 `gdcmWarningMacro`

```
#define gdcmWarningMacro(
    msg)
```

Value:

```
{
    if( gdcm::Trace::GetWarningFlag\(\) )
    {
        std::ostringstream osmacro;
        osmacro << "Warning: In " << \_\_FILE\_\_ ", line " << \_\_LINE\_\_
            << ", function " << GDCM\_FUNCTION << "\n";
        << msg << "\n\n";
        std::ostream &_os = gdcm::Trace::GetWarningStream\(\);
        _os << osmacro.str() << std::endl;
    }
}
GDCM\_NOOP\_STATEMENT
```

Warning.

Parameters

msg	message part
-----	--------------

Referenced by [gdcm::DataSet::InsertDataElement\(\)](#), [gdcm::Item::Read\(\)](#), [gdcm::SequenceOfItems::Read\(\)](#), [gdcm::Fragment::ReadBacktrack\(\)](#), [gdcm::Fragment::ReadValue\(\)](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::OpenSSL7CryptographicMessageSyntax::SetPassword\(\)](#), and [gdcm::Item::Write\(\)](#).

13.80 gdcmTrace.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTRACE_H
00015 #define GDCMTRACE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmSystem.h"
00019
00020 #include <iosfwd>
00021 #include <cassert>
00022
00023 namespace gdcm
00024 {
00025
00041 class GDCM_EXPORT Trace
00042 {
00043 public :
00044     Trace();
00045     ~Trace();
00046
00049     static void SetStream(std::ostream &os);
00050     static std::ostream &GetStream();
00051
00053     static void SetDebugStream(std::ostream &os);
00054     static std::ostream &GetDebugStream();
00055
00057     static void SetWarningStream(std::ostream &os);
00058     static std::ostream &GetWarningStream();
00059
00061     static void SetErrorStream(std::ostream &os);
00062     static std::ostream &GetErrorStream();
00063
00066     static void SetStreamToFile( const char *filename );
00067
00069     static void SetDebug(bool debug);
00070     static void DebugOn();
00071     static void DebugOff();
00072     static bool GetDebugFlag();
00073
00075     static void SetWarning(bool debug);
00076     static void WarningOn();
00077     static void WarningOff();
00078     static bool GetWarningFlag();
00079
00081     static void SetError(bool debug);
00082     static void ErrorOn();
00083     static void ErrorOff();
00084     static bool GetErrorFlag();
00085
00086 protected:
00087 private:
00088 };
00089
00090 // Here we define function this is the only way to be able to pass
00091 // stuff with indirection like:
00092 // gdcmDebug( "my message:" « i « '\t' );
00093 // You cannot use function unless you use vnsprintf ...
00094
00095 // __FUNCTION__ is not always defined by preprocessor
00096 // In c++ we should use __PRETTY_FUNCTION__ instead...
00097 #ifdef GDCM_CXX_HAS_FUNCTION
00098 // Handle particular case for GNU C++ which also defines __PRETTY_FUNCTION__

```

```

00099 // which is a lot nice in C++
00100 #ifdef __BORLANDC__
00101 # define __FUNCTION__ __FUNC__
00102 #endif
00103 #ifdef __GNUC__
00104 # define GDCM_FUNCTION __PRETTY_FUNCTION__
00105 #else
00106 # define GDCM_FUNCTION __FUNCTION__
00107 #endif // __GNUC__
00108 #else
00109 # define GDCM_FUNCTION "<unknown>"
00110 #endif //GDCM_CXX_HAS_FUNCTION
00111
00116 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00117 #define gdcmDebugMacro(msg) GDCM_NOOP_STATEMENT
00118 #else
00119 #define gdcmDebugMacro(msg)
00120 {
00121     if( gdcm::Trace::GetDebugFlag() )
00122     {
00123         std::ostringstream osmacro;
00124         osmacro << "Debug: In " __FILE__ ", line " << __LINE__
00125             << ", function " << GDCM_FUNCTION << "\n"
00126             << "Last system error was: "
00127             << gdcm::System::GetLastSystemError() << "\n" << msg;
00128         std::ostream &_os = gdcm::Trace::GetDebugStream();
00129         _os << osmacro.str() << "\n\n" << std::endl;
00130     }
00131 }
00132 GDCM_NOOP_STATEMENT
00133 #endif //NDEBUG
00134
00139 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00140 #define gdcmWarningMacro(msg) GDCM_NOOP_STATEMENT
00141 #else
00142 #define gdcmWarningMacro(msg)
00143 {
00144     if( gdcm::Trace::GetWarningFlag() )
00145     {
00146         std::ostringstream osmacro;
00147         osmacro << "Warning: In " __FILE__ ", line " << __LINE__
00148             << ", function " << GDCM_FUNCTION << "\n"
00149             << msg << "\n\n";
00150         std::ostream &_os = gdcm::Trace::GetWarningStream();
00151         _os << osmacro.str() << std::endl;
00152     }
00153 }
00154 GDCM_NOOP_STATEMENT
00155 #endif //NDEBUG
00156
00162 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00163 #define gdcmErrorMacro(msg) GDCM_NOOP_STATEMENT
00164 #else
00165 #define gdcmErrorMacro(msg)
00166 {
00167     if( gdcm::Trace::GetErrorFlag() )
00168     {
00169         std::ostringstream osmacro;
00170         osmacro << "Error: In " __FILE__ ", line " << __LINE__
00171             << ", function " << GDCM_FUNCTION << "\n"
00172             << msg << "\n\n";
00173         std::ostream &_os = gdcm::Trace::GetErrorStream();
00174         _os << osmacro.str() << std::endl;
00175     }
00176 }
00177 GDCM_NOOP_STATEMENT
00178 #endif //NDEBUG
00179
00186 #if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
00187 #define gdcmAssertMacro(arg) GDCM_NOOP_STATEMENT
00188 #else
00189 #define gdcmAssertMacro(arg)
00190 {
00191     if( !(arg) )
00192     {
00193         std::ostringstream osmacro;
00194         osmacro << "Assert: In " __FILE__ ", line " << __LINE__
00195             << ", function " << GDCM_FUNCTION
00196             << "\n\n";
00197         std::ostream &_os = gdcm::Trace::GetErrorStream();
00198         _os << osmacro.str() << std::endl;

```

```

00199  assert ( arg );
00200  }
00201 }
00202 GDCM_NOOP_STATEMENT
00203 #endif //NDEBUG
00204
00211 #if defined(NDEBUG)
00212 // User asked for release compilation, but still need to report
00213 // if grave issue.
00214 #define gdcmAssertAlwaysMacro(arg) \
00215 {
00216     if( !(arg) )
00217     {
00218         std::ostringstream osmacro;
00219         osmacro << "Assert: In " __FILE__ ", line " << __LINE__
00220             << ", function " << GDCM_FUNCTION
00221             << "\n\n";
00222         throw osmacro.str();
00223     }
00224 }
00225 GDCM_NOOP_STATEMENT
00226 #else
00227 // Simply reproduce gdcmAssertMacro behavior:
00228 #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
00229 #endif //NDEBUG
00230
00231 } // end namespace gdcm
00232 //-----
00233 #endif //GDCMTRACE_H

```

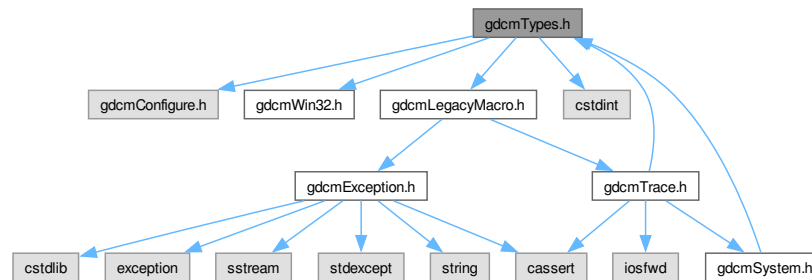
13.81 gdcmTypes.h File Reference

```

#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
#include <cstdlib>

```

Include dependency graph for gdcmTypes.h:



13.82 gdcmTypes.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library

```

```

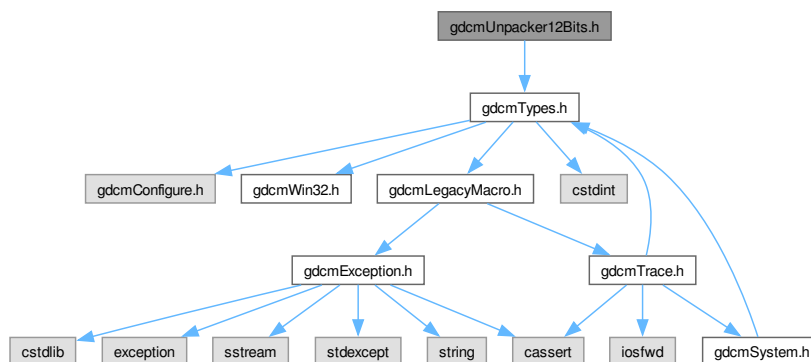
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMTYPES_H
00015 #define GDCMTYPES_H
00016
00017 #include "gdcmConfigure.h"
00018 #include "gdcmWin32.h"
00019 #include "gdcmLegacyMacro.h"
00020
00021 //-----
00022 #include <stdint>
00023
00024 //-----
00025 #endif //GDCMTYPES_H

```

13.83 gdcmUnpacker12Bits.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmUnpacker12Bits.h:



Classes

- class `gdcm::Unpacker12Bits`
Pack/Unpack 12 bits pixel into 16bits.

Namespaces

- namespace `gdcm`

13.84 gdcmUnpacker12Bits.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMUNPACKER12BITS_H
00015 #define GDCMUNPACKER12BITS_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00034 class GDCM_EXPORT Unpacker12Bits
00035 {
00036 public:
00040 static bool Pack(char *out, const char *in, size_t n);
00041
00045 static bool Unpack(char *out, const char *in, size_t n);
00046 };
00047
00048 } // end namespace gdcm
00049
00050 #endif //GDCMUNPACKER12BITS_H

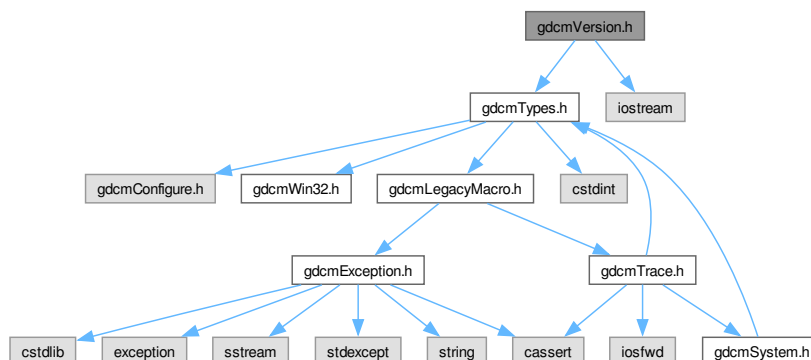
```

13.85 gdcmVersion.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmVersion.h:



Classes

- class [gdcm::Version](#)
major/minor and build version

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

13.86 gdcmVersion.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMVERSION_H
00015 #define GDCMVERSION_H
00016
00017 #include "gdcmTypes.h"
00018 #include <iostream>
00019
00020 namespace gdcm
00021 {
00022 //-----
00023 class GDCM_EXPORT Version
00024 {
00025 friend std::ostream& operator<<(std::ostream &_os, const Version &v);
00026 public :
00027 static const char *GetVersion();
00028 static int GetMajorVersion();
00029 static int GetMinorVersion();
00030 static int GetBuildVersion();
00031
00032 void Print(std::ostream &os = std::cout) const;
00033
00034 //protected:
00035 Version() = default;
00036 ~Version() = default;
00037 };
00038 //-----
00039 inline std::ostream& operator<<(std::ostream &os, const Version &v)
00040 {
00041 v.Print( os );
00042 return os;
00043 }
00044
00045 } // end namespace gdcm
00046 //-----
00047 #endif //GDCMVERSION_H

```


13.87 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define` [GDCM_EXPORT](#)

13.87.1 Macro Definition Documentation

13.87.1.1 GDCM_EXPORT

```
#define GDCM_EXPORT
```

Referenced by [gdcm::terminal::setAttribute\(\)](#), [gdcm::terminal::setbgcolor\(\)](#), [gdcm::terminal::setfgcolor\(\)](#), and [gdcm::terminal::setmode\(\)](#).

13.88 gdcmWin32.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMWIN32_H
00016 #define GDCMWIN32_H
00017
00018 #if !defined(GDCMTYPES_H)
00019 #error you need to include gdcmTypes.h instead
00020 #endif
00021 //-----
00022 // http://gcc.gnu.org/wiki/Visibility
00023 #if defined(_WIN32) && defined(GDCM_BUILD_SHARED_LIBS)
00024 #if (defined(gdcmCommon_EXPORTS) || defined(gdcmDICT_EXPORTS) || defined(gdcmDSED_EXPORTS) ||
00025      defined(gdcmIOD_EXPORTS) || defined(gdcmMSFF_EXPORTS) || defined(gdcmMEXD_EXPORTS)||
00026      defined(_gdcmSwig_EXPORTS)) || defined(vtkgdcm_EXPORTS)
00025      #define GDCM_EXPORT __declspec( dllexport )
00026 #else
00027      #define GDCM_EXPORT __declspec( dllimport )
00028 #endif
00029 #else
00030 #if __GNUC__ >= 4 && defined(GDCM_BUILD_SHARED_LIBS)
00031      #define GDCM_EXPORT __attribute__((visibility ("default")))
```

```

00032  #define GDCM_LOCAL  __attribute__((visibility ("hidden")))
00033  #else
00034  #define GDCM_EXPORT
00035  #endif
00036 #endif
00037
00038 #if defined(GDCM_OVERRIDE_BROKEN_IMPLEMENTATION) && !defined(GDCM_FORCE_EXPORT)
00039 #undef GDCM_EXPORT
00040 #define GDCM_EXPORT
00041 #endif
00042
00043 // In VTK 4.2 vtkWrapPython does not like anything other than VTK_*EXPORT
00044 // [ 86%] Generating vtkGDCMImageReaderPython.cxx
00045 // syntax error
00046 // *** SYNTAX ERROR found in parsing the header file
//usr/local/src/gdcm2/tags/gdcm-2-0-11/Utilities/VTK/vtkGDCMImageReader.h before line 128***
00047 // make[2]: *** [Utilities/VTK/vtkGDCMImageReaderPython.cxx] Error 1
00048 // make[1]: *** [Utilities/VTK/CMakeFiles/vtkgdcmPythonD.dir/all] Error 2
00049 // make: *** [all] Error 2
00050
00051 #if defined(VTK_MAJOR_VERSION) && ( VTK_MAJOR_VERSION == 4 )
00052 #undef VTK_EXPORT
00053 #define VTK_EXPORT GDCM_EXPORT
00054 #endif
00055
00056 //-----
00057 //This is needed when compiling in debug mode
00058 #ifdef _MSC_VER
00059 // to allow construct such as: std::numeric_limits<int>::max() we need the following:
00060 // warning C4003: not enough actual parameters for macro 'max'
00061 #ifndef NOMINMAX
00062 #define NOMINMAX
00063 #endif
00064 # pragma warning ( default : 4263 ) /* no override, call convention differs */
00065 // 'identifier' : class 'type' needs to have dll-interface to be used by
00066 // clients of class 'type2'
00067 #pragma warning ( disable : 4251 )
00068 // non dll-interface class 'type' used as base for dll-interface class 'type2'
00069 #pragma warning ( disable : 4275 )
00070 // 'identifier' : identifier was truncated to 'number' characters in the
00071 // debug information
00072 #pragma warning ( disable : 4786 )
00073 // 'identifier' : decorated name length exceeded, name was truncated
00074 #pragma warning ( disable : 4503 )
00075 #endif // _MSC_VER
00076
00077 //-----
00078 #endif //GDCMWIN32_H

```

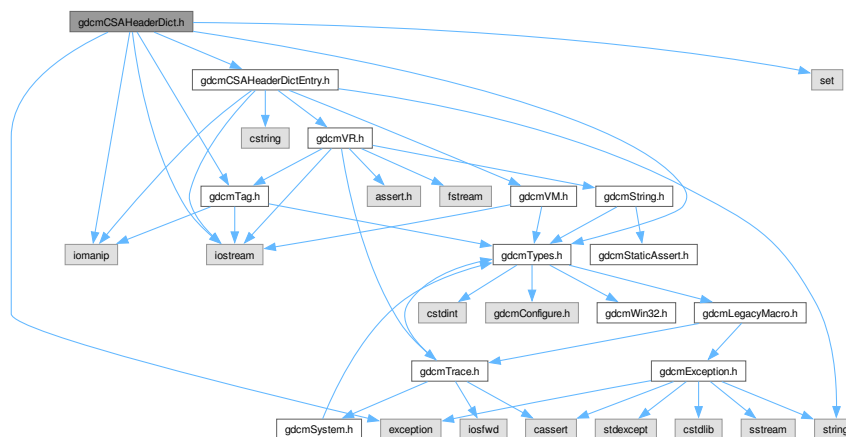
13.89 gdcmCSAHeaderDict.h File Reference

```

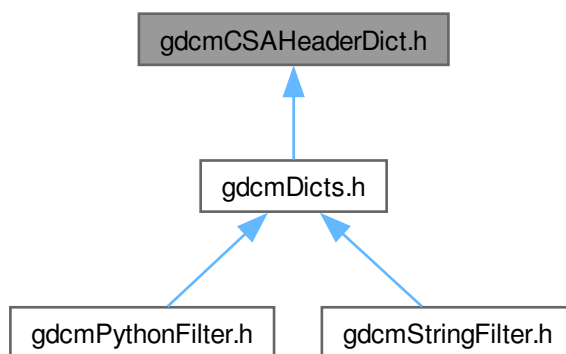
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmCSAHeaderDictEntry.h"
#include <iostream>
#include <iomanip>
#include <set>
#include <exception>

```

Include dependency graph for gdcmCSAHeaderDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::CSAHeaderDict](#)
Class to represent a map of [CSAHeaderDictEntry](#).
- class [gdcm::CSAHeaderDictException](#)

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

13.90 gdcmCSAHeaderDict.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCSAHEADERDICT_H
00015 #define GDCMCSAHEADERDICT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmCSAHeaderDictEntry.h"
00020
00021 #include <iostream>
00022 #include <iomanip>
00023 #include <set>
00024 #include <exception>
00025
00026 namespace gdcm
00027 {
00028
00029 class GDCM_EXPORT CSAHeaderDictException : public std::exception {};
00030
00031 class GDCM_EXPORT CSAHeaderDict
00032 {
00033 public:
00034     typedef std::set<CSAHeaderDictEntry> MapCSAHeaderDictEntry;
00035     typedef MapCSAHeaderDictEntry::iterator Iterator;
00036     typedef MapCSAHeaderDictEntry::const_iterator ConstIterator;
00037     //static CSAHeaderDictEntry GroupLengthCSAHeaderDictEntry; // = CSAHeaderDictEntry("Group
00038     Length",VR::UL,VM::VM1);
00039
00040     CSAHeaderDict():CSAHeaderDictInternal() {
00041         gdcm_assert( CSAHeaderDictInternal.empty() );
00042     }
00043     CSAHeaderDict &operator=(const CSAHeaderDict &_val) = delete;
00044     CSAHeaderDict(const CSAHeaderDict &_val) = delete;
00045
00046     friend std::ostream& operator<<(std::ostream& _os, const CSAHeaderDict &_val);
00047
00048     ConstIterator Begin() const { return CSAHeaderDictInternal.begin(); }
00049     ConstIterator End() const { return CSAHeaderDictInternal.end(); }
00050
00051     bool IsEmpty() const { return CSAHeaderDictInternal.empty(); }
00052     void AddCSAHeaderDictEntry(const CSAHeaderDictEntry &de)
00053     {
00054         #ifndef NDEBUG
00055             MapCSAHeaderDictEntry::size_type s = CSAHeaderDictInternal.size();
00056             CSAHeaderDictInternal.insert( de );
00057             #ifndef NDEBUG
00058                 gdcm_assert( s < CSAHeaderDictInternal.size() );
00059             #endif
00060         }
00061     }
00062
00063     const CSAHeaderDictEntry &GetCSAHeaderDictEntry(const char *name) const
00064     {
00065         MapCSAHeaderDictEntry::const_iterator it = CSAHeaderDictInternal.find( name );

```

```

00068     if( it != CSAHeaderDictInternal.end() )
00069     {
00070         return *it;
00071     }
00072     throw CSAHeaderDictException();
00073 }
00074
00075 protected:
00076 friend class Dicts;
00077 void LoadDefault();
00078
00079 private:
00080
00081 MapCSAHeaderDictEntry CSAHeaderDictInternal;
00082 };
00083 //-----
00084 inline std::ostream& operator<<(std::ostream& os, const CSAHeaderDict &val)
00085 {
00086     CSAHeaderDict::MapCSAHeaderDictEntry::const_iterator it = val.CSAHeaderDictInternal.begin();
00087     for(; it != val.CSAHeaderDictInternal.end(); ++it)
00088     {
00089         const CSAHeaderDictEntry &de = *it;
00090         os << de << '\n';
00091     }
00092
00093     return os;
00094 }
00095 }
00096
00097 } // end namespace gdcml
00098 } // end namespace gdcml
00099
00100 #endif //GDCMLCSAHEADERDICT_H

```

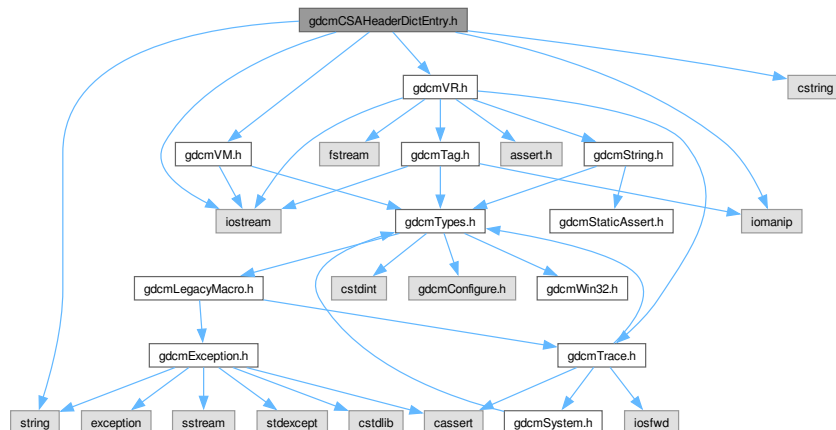
13.91 gdcmlCSAHeaderDictEntry.h File Reference

```

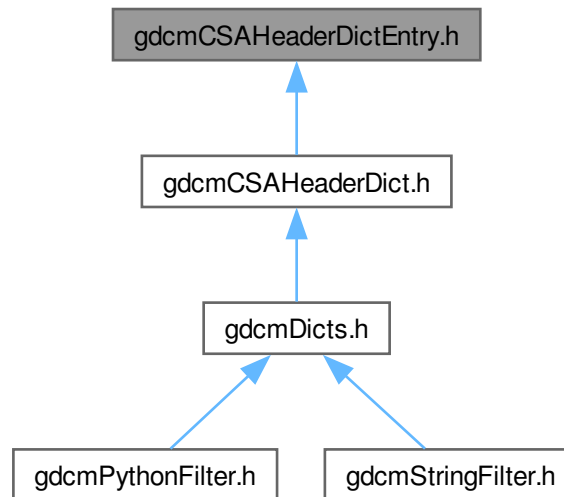
#include "gdcmlVR.h"
#include "gdcmlVM.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <cstring>

```

Include dependency graph for gdcmlCSAHeaderDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::CSAHeaderDictEntry](#)
Class to represent an Entry in the [Dict](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

13.92 gdcmCSAHeaderDictEntry.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008

```

```

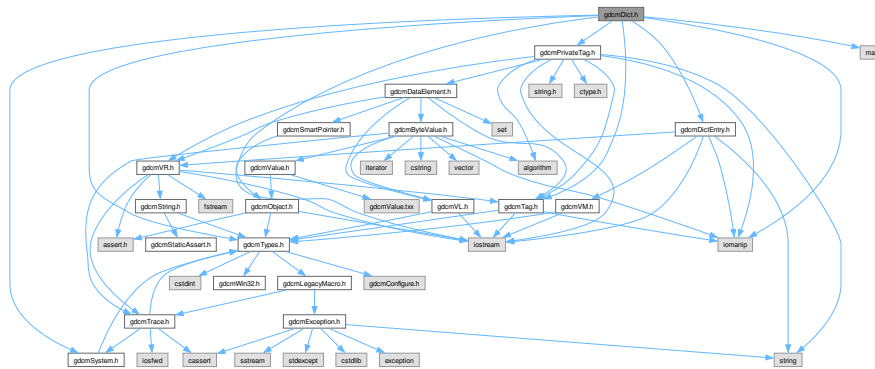
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014 #ifndef GDCMCSAHEADERDICTENTRY_H
00015 #define GDCMCSAHEADERDICTENTRY_H
00016
00017 #include "gdcmVR.h"
00018 #include "gdcmVM.h"
00019
00020 #include <string>
00021 #include <iostream>
00022 #include <iomanip>
00023
00024 #include <cstring>
00025
00026 namespace gdcm
00027 {
00028     class GDCM_EXPORT CSAHeaderDictEntry
00029     {
00030     public:
00031         CSAHeaderDictEntry(const char *name = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0, const char *desc
= ""):Name(name),ValueRepresentation(vr),ValueMultiplicity(vm),Description(desc) {
00032         }
00033
00034         friend std::ostream& operator<(std::ostream& _os, const CSAHeaderDictEntry &_val);
00035
00036         const VR &GetVR() const { return ValueRepresentation; }
00037         void SetVR(const VR & vr) { ValueRepresentation = vr; }
00038
00039         const VM &GetVM() const { return ValueMultiplicity; }
00040         void SetVM(VM const & vm) { ValueMultiplicity = vm; }
00041
00042         const char *GetName() const { return Name.c_str(); }
00043         void SetName(const char* name) { Name = name; }
00044
00045         const char *GetDescription() const { return Description.c_str(); }
00046         void SetDescription(const char* desc) { Description = desc; }
00047
00048         bool operator<(const CSAHeaderDictEntry &entry) const
00049         {
00050             return strcmp(GetName(),entry.GetName()) < 0;
00051         }
00052
00053     private:
00054         std::string Name;
00055         VR ValueRepresentation;
00056         VM ValueMultiplicity;
00057         std::string Description;
00058         std::string Type; // TODO
00059     };
00060
00061 //-----
00062 inline std::ostream& operator<(std::ostream& os, const CSAHeaderDictEntry &val)
00063 {
00064     if( val.Name.empty() )
00065     {
00066         os << "[No name]";
00067     }
00068     else
00069     {
00070         os << val.Name;
00071     }
00072     os << "\t" << val.ValueRepresentation << "\t" << val.ValueMultiplicity;
00073     if( !val.Description.empty() )
00074     {
00075         os << "\t" << val.Description;
00076     }
00077     return os;
00078 }
00079
00080 } // end namespace gdcm
00081
00082 #endif //GDCMCSAHEADERDICTENTRY_H

```

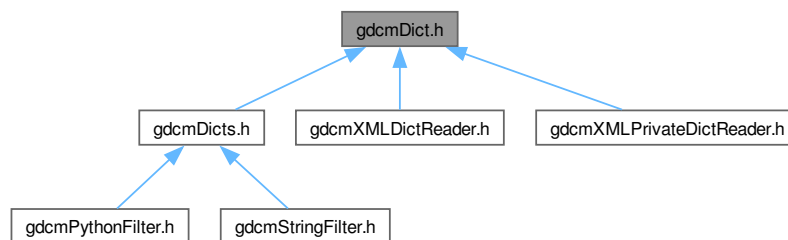
13.93 gdcmDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
#include <map>
```

Include dependency graph for gdcmDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Dict](#)
Class to represent a map of [DictEntry](#).
- class [gdcm::PrivateDict](#)
Private [Dict](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)`

13.94 gdcmDict.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDICT_H
00015 #define GDCMDICT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmPrivateTag.h"
00020 #include "gdcmDictEntry.h"
00021 #include "gdcmSystem.h"
00022
00023 #include <iostream>
00024 #include <iomanip>
00025 #include <map>
00026
00027 /*
00028  * FIXME / TODO
00029  * I need to seriously rewrite this mess. a class template should work for both a public
00030  * and a private dict
00031  */
00032
00033 namespace gdcm
00034 {
00035  // Data Element Tag
00036  class GDCM_EXPORT Dict
00037  {
00038  public:
00039      typedef std::map<Tag, DictEntry> MapDictEntry;
00040      typedef MapDictEntry::iterator Iterator;
00041      typedef MapDictEntry::const_iterator ConstIterator;
00042      //static DictEntry GroupLengthDictEntry; // = DictEntry("Group Length",VR::UL,VM::VM1);
00043
00044      Dict():DictInternal() {
00045          gdcm_assert( DictInternal.empty() );
00046      }
00047      Dict &operator=(const Dict &_val) = delete;
00048      Dict(const Dict &_val) = delete;
00049
00050      friend std::ostream& operator<<(std::ostream& _os, const Dict &_val);
00051
00052      ConstIterator Begin() const { return DictInternal.begin(); }
00053      ConstIterator End() const { return DictInternal.end(); }
00054  }
00055
00056
00057
00058
00059
00060
00061
00062
00063

```

```

00064 bool IsEmpty() const { return DictInternal.empty(); }
00065 void AddDictEntry(const Tag &tag, const DictEntry &de)
00066 {
00067 #ifndef NDEBUB
00068     MapDictEntry::size_type s = DictInternal.size();
00069 #endif
00070     DictInternal.insert(
00071         MapDictEntry::value_type(tag, de));
00072 #ifndef NDEBUB
00073     gdcmm_assert( s < DictInternal.size() );
00074 #endif
00075 }
00076
00077 const DictEntry &GetDictEntry(const Tag &tag) const
00078 {
00079     MapDictEntry::const_iterator it =
00080         DictInternal.find(tag);
00081     if (it == DictInternal.end())
00082     {
00083 #ifdef UNKNOWNPUBLICTAG
00084         // test.acr
00085         if( tag != Tag(0x28,0x15)
00086             && tag != Tag(0x28,0x16)
00087             && tag != Tag(0x28,0x199)
00088             // gdcmmData/TherapysGDCM1.dcm
00089             && tag != Tag(0x20,0x1)
00090             // gdcmmData/0019004_Baseline_IMG1.dcm
00091             && tag != Tag(0x8348,0x339)
00092             && tag != Tag(0xb5e8,0x338)
00093             // gdcmmData/dicomdir_Acusson_WithPrivate_WithSR
00094             && tag != Tag(0x40,0xa125)
00095         )
00096         {
00097             gdcmm_assert( 0 && "Impossible" );
00098         }
00099 #endif
00100         it = DictInternal.find( Tag(0xffff,0xffff) );
00101         return it->second;
00102     }
00103     gdcmm_assert( DictInternal.count(tag) == 1 );
00104     return it->second;
00105 }
00106
00107 const char *GetKeywordFromTag(Tag const & tag) const
00108 {
00109     MapDictEntry::const_iterator it =
00110         DictInternal.find(tag);
00111     if (it == DictInternal.end())
00112     {
00113         return nullptr;
00114     }
00115     gdcmm_assert( DictInternal.count(tag) == 1 );
00116     return it->second.GetKeyword();
00117 }
00118
00119 const DictEntry &GetDictEntryByKeyword(const char *keyword, Tag & tag) const
00120 {
00121     MapDictEntry::const_iterator it =
00122         DictInternal.begin();
00123     if( keyword )
00124     {
00125         for( it != DictInternal.end(); ++it )
00126         {
00127             if( strcmp( keyword, it->second.GetKeyword() ) == 0 )
00128             {
00129                 // Found a match !
00130                 tag = it->first;
00131                 break;
00132             }
00133         }
00134     }
00135     else
00136     {
00137         it = DictInternal.end();
00138     }
00139     if (it == DictInternal.end())
00140     {
00141         tag = Tag(0xffff,0xffff);
00142         it = DictInternal.find( tag );
00143         return it->second;
00144     }
00145 }

```

```

00150     gdcml_assert( DictInternal.count(tag) == 1 );
00151     return it->second;
00152 }
00153
00154 const DictEntry &GetDictEntryByName(const char *name, Tag & tag) const
00155 {
00156     MapDictEntry::const_iterator it =
00157         DictInternal.begin();
00158     if( name )
00159     {
00160         for( ; it != DictInternal.end(); ++it )
00161         {
00162             if( strcmp( name, it->second.GetName() ) == 0 )
00163             {
00164                 // Found a match !
00165                 tag = it->first;
00166                 break;
00167             }
00168         }
00169     }
00170     else
00171     {
00172         it = DictInternal.end();
00173     }
00174     if (it == DictInternal.end())
00175     {
00176         tag = Tag(0xffff,0xffff);
00177         it = DictInternal.find( tag );
00178         return it->second;
00179     }
00180     gdcml_assert( DictInternal.count(tag) == 1 );
00181     return it->second;
00182 }
00183
00184 protected:
00185 friend class Dicts;
00186 void LoadDefault();
00187
00188 private:
00189 MapDictEntry DictInternal;
00190 };
00191 //-----
00192 inline std::ostream& operator<<(std::ostream& os, const Dict &val)
00193 {
00194     Dict::MapDictEntry::const_iterator it = val.DictInternal.begin();
00195     for(;it != val.DictInternal.end(); ++it)
00196     {
00197         const Tag &t = it->first;
00198         const DictEntry &de = it->second;
00199         os << t << " " << de << "\n";
00200     }
00201     return os;
00202 }
00203
00204 // TODO
00205 // For private dict, element < 0x10 should automatically defined:
00206 // Name = "Private Creator"
00207 // ValueRepresentation = LO
00208 // ValueMultiplicity = 1
00209 // Owner = ""
00210
00211 class GDCM_EXPORT PrivateDict
00212 {
00213     typedef std::map<PrivateTag, DictEntry> MapDictEntry;
00214     friend std::ostream& operator<<(std::ostream& os, const PrivateDict &val);
00215 public:
00216     PrivateDict() = default;
00217     ~PrivateDict() = default;
00218     void AddDictEntry(const PrivateTag &tag, const DictEntry &de)
00219     {
00220         #ifndef NDEBUG
00221             MapDictEntry::size_type s = DictInternal.size();
00222             DictInternal.insert(
00223                 MapDictEntry::value_type(tag, de));
00224             // The following code should only be used when manually constructing a Private.xml file by hand
00225             // it will get rid of VR::UN duplicate (ie. if a VR != VR::Un can be found)
00226             #if defined(NDEBUG) && 0
00227                 if( s == DictInternal.size() )
00228                 {

```

```

00237     MapDictEntry::iterator it =
00238         DictInternal.find(tag);
00239     gdcmm_assert( it != DictInternal.end() );
00240     DictEntry &duplicate = it->second;
00241     gdcmm_assert( de.GetVR() == VR::UN || duplicate.GetVR() == VR::UN );
00242     gdcmm_assert( de.GetVR() != duplicate.GetVR() );
00243     if( duplicate.GetVR() == VR::UN )
00244     {
00245         gdcmm_assert( de.GetVR() != VR::UN );
00246         duplicate.SetVR( de.GetVR() );
00247         duplicate.SetVM( de.GetVM() );
00248         gdcmm_assert( GetDictEntry(tag).GetVR() != VR::UN );
00249         gdcmm_assert( GetDictEntry(tag).GetVR() == de.GetVR() );
00250         gdcmm_assert( GetDictEntry(tag).GetVM() == de.GetVM() );
00251     }
00252     return;
00253 }
00254 #endif
00255 #ifndef NDEBUG
00256     gdcmm_assert( s < DictInternal.size() /*&& std::cout << tag << ", " << de << std::endl*/ );
00257 #endif
00258 }
00261 bool RemoveDictEntry(const PrivateTag &tag)
00262 {
00263     MapDictEntry::size_type s =
00264         DictInternal.erase(tag);
00265     gdcmm_assert( s == 1 || s == 0 );
00266     return s == 1;
00267 }
00268 bool FindDictEntry(const PrivateTag &tag) const
00269 {
00270     MapDictEntry::const_iterator it =
00271         DictInternal.find(tag);
00272     if (it == DictInternal.end())
00273     {
00274         return false;
00275     }
00276     return true;
00277 }
00278 const DictEntry &GetDictEntry(const PrivateTag &tag) const
00279 {
00280     // if 0x10 -> return Private Creator
00281     MapDictEntry::const_iterator it =
00282         DictInternal.find(tag);
00283     if (it == DictInternal.end())
00284     {
00285         //gdcmm_assert( 0 && "Impossible" );
00286         it = DictInternal.find( PrivateTag(0xffff,0xffff,"GDCM Private Sentinel" ) );
00287         assert( it != DictInternal.end());
00288         return it->second;
00289     }
00290     gdcmm_assert( DictInternal.count(tag) == 1 );
00291     return it->second;
00292 }
00293
00294
00295 void PrintXML() const
00296 {
00297     MapDictEntry::const_iterator it = DictInternal.begin();
00298     std::cout << "<dict edition=\"2008\">\n";
00299     for(;it != DictInternal.end(); ++it)
00300     {
00301         const PrivateTag &t = it->first;
00302         const DictEntry &de = it->second;
00303         std::cout << " <entry group=\"\" <std::hex < std::setw(4)
00304             << std::setfill('0') << t.GetGroup() << "\" <
00305             << " element=\"xx\" < std::setw(2) < std::setfill('0') << t.GetElement() << "\" <
00306             << " de.GetVR() < \"\" vm=\"\" < de.GetVM() < \"\" owner=\"\"
00307             << t.GetOwner();
00308         const char *name = de.GetName();
00309         if( *name == 0 )
00310         {
00311             std::cout << "\"/>\n";
00312         }
00313         else
00314         {
00315             std::cout << "\" name=\"\" < de.GetName() < "\"/>\n";
00316         }
00317     }
00318     std::cout << "</dict>\n";
00319 }

```

```

00320
00321 bool IsEmpty() const { return DictInternal.empty(); }
00322 protected:
00323 friend class Dicts;
00324 void LoadDefault();
00325
00326 private:
00327 PrivateDict &operator=(const PrivateDict &_val) = delete;
00328 PrivateDict(const PrivateDict &_val) = delete;
00329
00330 MapDictEntry DictInternal;
00331 };
00332 //-----
00333 inline std::ostream& operator<<(std::ostream& os, const PrivateDict &val)
00334 {
00335     PrivateDict::MapDictEntry::const_iterator it = val.DictInternal.begin();
00336     for(; it != val.DictInternal.end(); ++it)
00337     {
00338         const PrivateTag &t = it->first;
00339         const DictEntry &de = it->second;
00340         os << t << " " << de << '\n';
00341     }
00342
00343     return os;
00344 }
00345
00346 } // end namespace gdcm
00347
00348 #endif //GDCMDICT_H

```

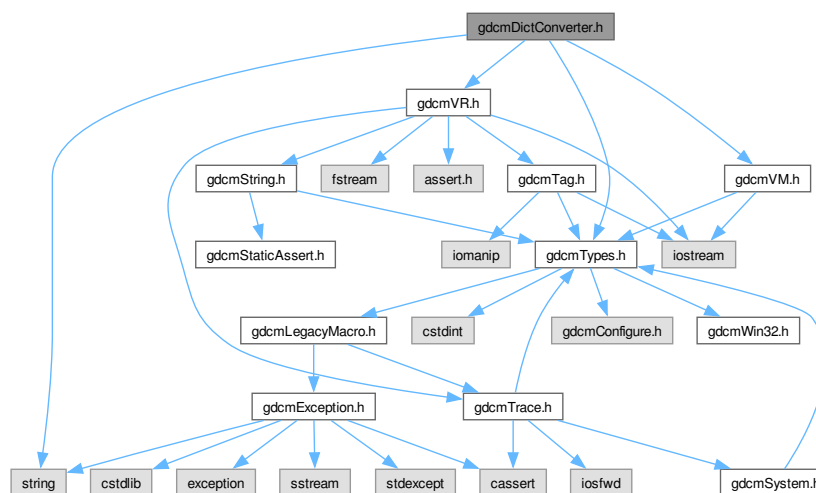
13.95 gdcmDictConverter.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>

```

Include dependency graph for gdcmDictConverter.h:



Classes

- class [gdcm::DictConverter](#)

Class to convert a .dic file into something else:

Namespaces

- namespace `gdcm`

13.96 `gdcmDictConverter.h`

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMDICTCONVERTER_H
00016 #define GDCMDICTCONVERTER_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmVM.h"
00021
00022 #include <string>
00023
00024 namespace gdcm
00025 {
00026
00027 class DictConverterInternal;
00036 class GDCM_EXPORT DictConverter
00037 {
00038 public:
00039 DictConverter();
00040 ~DictConverter();
00041 void SetInputFileName(const char* filename);
00042 const std::string &GetInputFilename() const;
00043 void SetOutputFileName(const char* filename);
00044 const std::string &GetOutputFilename() const;
00045
00046 int GetOutputType() const {
00047     return OutputType;
00048 }
00049 void SetOutputType(int type) {
00050     OutputType = type;
00051 }
00052 const std::string &GetDictName() const;
00053 void SetDictName(const char *name);
00054
00055 void Convert();
00056
00057 // Leaving them public for now. Not really user oriented but may be
00058 // useful
00059 static bool ReadVR(const char *raw, VR::VRType &type);
00060 static bool ReadVM(const char *raw, VM::VMType &type);
00061 static bool Readuint16(const char *raw, uint16_t &ov);
00062
00063 enum OutputTypes {
00064     DICT_DEFAULT = 0,
00065     DICT_DEBUG,
00066     DICT_XML
00067 };
00068
00069 protected:
00070 void WriteHeader();
00071 void WriteFooter();
00072 bool ConvertToXML(const char *raw, std::string &cxx);
00073 bool ConvertToCXX(const char *raw, std::string &cxx);

```

```

00074 void AddGroupLength();
00075
00076 private:
00077 DictConverterInternal *Internal;
00078
00079 int OutputType;
00080 };
00081
00082 } // end namespace gdcm
00083
00084 #endif //GDCMDICTCONVERTER_H

```

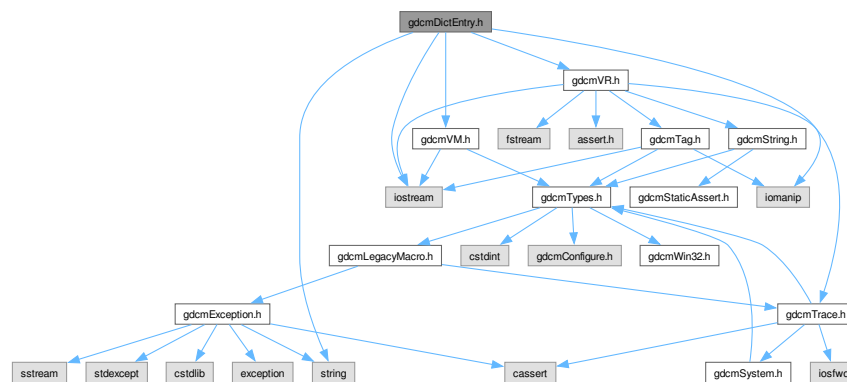
13.97 gdcmDictEntry.h File Reference

```

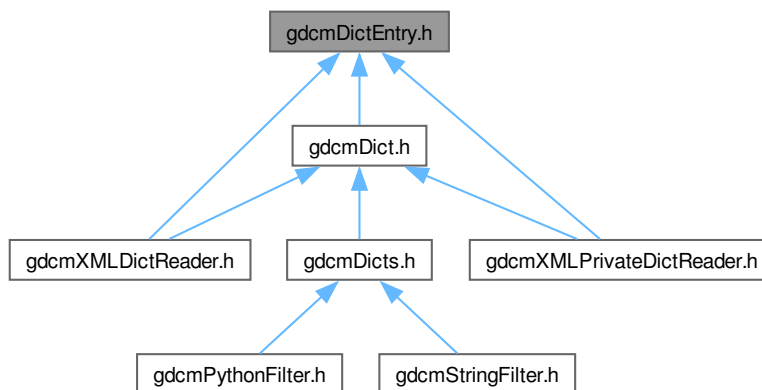
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>

```

Include dependency graph for gdcmDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::DictEntry](#)
Class to represent an Entry in the [Dict](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)`

13.98 gdcmDictEntry.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDICTENTRY_H

```



```

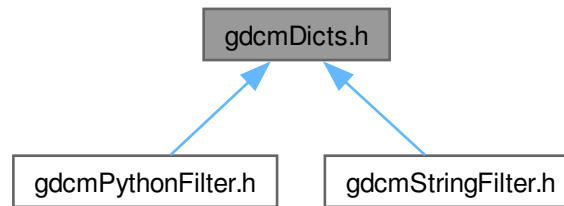
00015 #define GDCMDICTENTRY_H
00016
00017 #include "gdcmVR.h"
00018 #include "gdcmVM.h"
00019
00020 #include <string>
00021 #include <iostream>
00022 #include <iomanip>
00023
00024 namespace gdcm
00025 {
00036 class GDCM_EXPORT DictEntry
00037 {
00038 public:
00039     DictEntry(const char *name = "", const char *keyword = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0,
00040         bool ret = false):
00041         Name(name),
00042         Keyword(keyword),
00043         ValueRepresentation(vr),
00044         ValueMultiplicity(vm),
00045         Retired(ret),
00046         GroupXX(false),
00047         ElementXX(false)
00048     {
00049
00050     friend std::ostream& operator<<(std::ostream& _os, const DictEntry &_val);
00051
00052     const VR &GetVR() const { return ValueRepresentation; }
00053     void SetVR(const VR &vr) { ValueRepresentation = vr; }
00054     // bool IsValid() const { return ValueRepresentation != VR::VR_END; }
00055     // !Name.empty() /*&& ValueRepresentation && ValueMultiplicity*/; }
00056
00057     const VM &GetVM() const { return ValueMultiplicity; }
00058     void SetVM(VM const &vm) { ValueMultiplicity = vm; }
00059
00060     const char *GetName() const { return Name.c_str(); }
00061     void SetName(const char* name) { Name = name; }
00062
00063     const char *GetKeyword() const { return Keyword.c_str(); }
00064     void SetKeyword(const char* keyword) { Keyword = keyword; }
00065
00066     bool GetRetired() const { return Retired; }
00067     void SetRetired(bool retired) { Retired = retired; }
00068
00069     // <entry group="50xx" element="0005" vr="US" vm="1" retired="true" version="3">
00070     void SetGroupXX(bool v) { GroupXX = v; }
00071
00072     // <entry group="0020" element="31xx" vr="CS" vm="1-n" retired="true" version="2">
00073     void SetElementXX(bool v) { ElementXX = v; }
00074
00075     bool IsUnique() const { return ElementXX == false && GroupXX == false; }
00076
00077 private:
00078     //
00079     friend class Dict;
00080     static bool CheckKeywordAgainstName(const char *name, const char *keyword);
00081
00082 private:
00083     std::string Name;
00084     std::string Keyword;
00085     VR ValueRepresentation;
00086     VM ValueMultiplicity;
00087     bool Retired : 1;
00088     bool GroupXX : 1;
00089     bool ElementXX : 1;
00090 };
00091
00092 #if 0
00093 class GDCM_EXPORT PrivateDictEntry : public DictEntry
00094 {
00095 public:
00096     PrivateDictEntry(const char *name = "", VR::VRType const &vr = VR::INVALID, VM::VMType const &vm = VM::VM0 ,
00097         bool ret = false, const char *owner = ""):DictEntry(name,vr,vm,ret),Owner(owner) {}
00098     PrivateDictEntry(const char *name, const char *vr, const char *vm):DictEntry(name,vr,vm) {}
00099
00100     const char *GetOwner() const { return Owner.c_str(); }
00101     void SetOwner(const char *owner) { Owner = owner; }
00102
00103 private:
00104     // SIEMENS MED, GEMS_PETD_01 ...

```

13.99 gdcDicts.h File Reference

[illegible]

This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Dicts](#)
Class to manipulate the sum of knowledge (all the dict user load).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)`

13.100 gdcmDicts.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMDICTS_H
00015  #define GDCMDICTS_H
00016  00017  #include "gdcmDict.h"
00018  #include "gdcmCSAHeaderDict.h"
00019  00020  #include <string>
00021

```

```

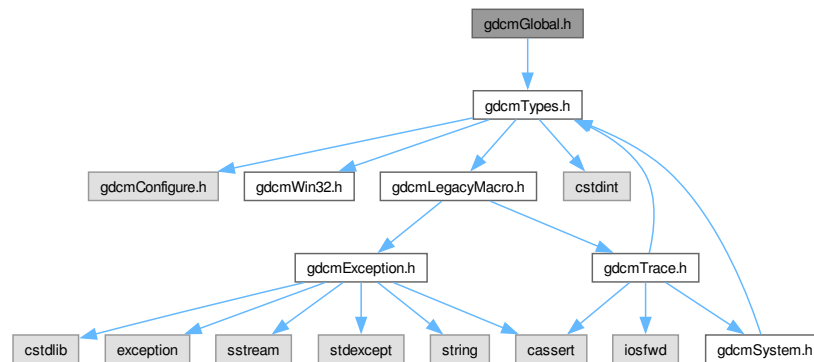
00022 namespace gdcm
00023 {
00028 class GDCM_EXPORT Dicts
00029 {
00030 friend std::ostream& operator«(std::ostream &_os, const Dicts &d);
00031 public:
00032 Dicts();
00033 ~Dicts();
00034 Dicts &operator=(const Dicts &_val) = delete;
00035 Dicts(const Dicts &_val) = delete;
00036
00040 // DataSet::GetPrivateCreator
00042 const DictEntry &GetDictEntry(const Tag& tag, const char *owner = nullptr) const;
00043
00044 const DictEntry &GetDictEntry(const PrivateTag& tag) const;
00045
00046 //enum PublicTypes {
00047 // DICOMV3_DICT,
00048 // ACRNEMA_DICT,
00049 // NIH_DICT
00050 //};
00051 const Dict &GetPublicDict() const;
00052
00053 const PrivateDict &GetPrivateDict() const;
00054 PrivateDict &GetPrivateDict();
00055
00056 const CSAHeaderDict &GetCSAHeaderDict() const;
00057
00058 bool IsEmpty() const { return GetPublicDict().IsEmpty(); }
00059
00060 protected:
00061 typedef enum {
00062 PHILIPS,
00063 GEMS,
00064 SIEMENS
00065 // ...
00066 } ConstructorType;
00067 static const char *GetConstructorString(ConstructorType type);
00068
00069 friend class Global;
00070 void LoadDefaults();
00071
00072 private:
00073 // Public dict:
00074 Dict PublicDict;
00075
00076 // Private Dicts:
00077 PrivateDict ShadowDict;
00078
00079 CSAHeaderDict CSADict;
00080 };
00081 //-----
00082 inline std::ostream& operator«(std::ostream &os, const Dicts &d)
00083 {
00084 (void)d;
00085 return os;
00086 }
00087
00088
00089 } // end namespace gdcm
00090
00091 #endif //GDCMDICTS_H

```

13.101 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



Classes

- class `gdcm::Global`
`Global`.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)`

Variables

- static `Global` `gdcm::GlobalInstance`

13.102 gdcmGlobal.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 // Implementation detail was shamelessly borrowed from the VTK excellent
00015 // implementation of debug leak manager singleton:
00016
00017  /*=====
00018  Program: Visualization Toolkit
00019  Module: $RCSfile: vtkDebugLeaks.cxx,v $
00020
00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 #ifndef GDCMGLOBAL_H
00031 #define GDCMGLOBAL_H
00032
00033 #include "gdcmTypes.h"
00034
00035 namespace gdcm
00036 {
00037 class GlobalInternal;
00038 class Dicts;
00039 class Defs;
00040 class GDCM_EXPORT Global // why expose the symbol I think I only need to expose the instance...
00041 {
00042 friend std::ostream& operator<<(std::ostream &_os, const Global &g);
00043 public:
00044 Global();
00045 ~Global();
00046 Global &operator=(const Global &_val) = delete;
00047 Global(const Global &_val) = delete;
00048
00049 Dicts const &GetDicts() const;
00050 Dicts &GetDicts();
00051
00052 Defs const &GetDefs() const;
00053
00054 static Global& GetInstance();
00055
00056 bool LoadResourcesFiles();
00057
00058 bool Append(const char *path);
00059
00060 bool Prepend(const char *path);
00061
00062 protected:
00063 const char *Locate(const char *resfile) const;
00064
00065 private:
00066 // PIMPL:
00067 // but we could have also directly exposed a Dicts *Internals;
00068 static GlobalInternal *Internals;
00069 };
00070 //-----
00071 inline std::ostream& operator<<(std::ostream &os, const Global &g)

```

```

00094 {
00095     (void)g;
00096     return os;
00097 }
00098
00099 // This instance will show up in any translation unit that uses
00100 // Global or that has a singleton. It will make sure
00101 // Global is initialized before it is used and is the last
00102 // static object destroyed.
00103 static Global GlobalInstance;
00104
00105 } // end namespace gdcm
00106
00107 #endif //GDCMGLOBAL_H

```

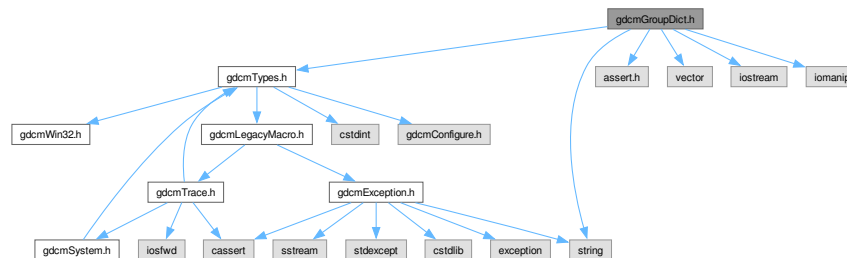
13.103 gdcmGroupDict.h File Reference

```

#include "gdcmTypes.h"
#include <assert.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>

```

Include dependency graph for gdcmGroupDict.h:



Classes

- class [gdcm::GroupDict](#)
Class to represent the mapping from group number to its abbreviation and name.

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)`

13.104 gdcmGroupDict.h

[Go to the documentation of this file.](#)

```

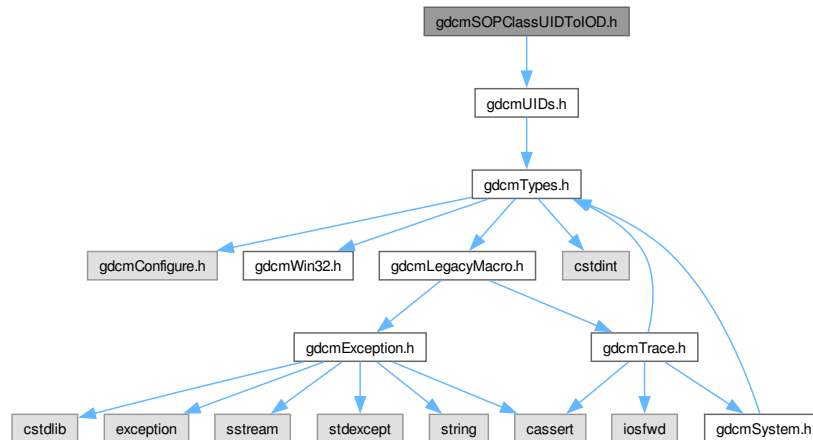
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMGROUPDICT_H
00016 #define GDCMGROUPDICT_H
00017
00018 #include "gdcmTypes.h"
00019
00020 #include <assert.h>
00021 #include <vector>
00022 #include <string>
00023 #include <iostream>
00024 #include <iomanip>
00025
00026 namespace gdcm
00027 {
00033 class GDCM_EXPORT GroupDict
00034 {
00035 public:
00036 typedef std::vector<std::string> GroupStringVector;
00037 GroupDict() { FillDefaultGroupName(); }
00038 ~GroupDict() = default;
00039
00040 friend std::ostream& operator<<(std::ostream& __os, const GroupDict &__val);
00041
00042 size_t Size() const
00043 {
00044     gdcm_assert( Names.size() == Abbreviations.size() );
00045     return Names.size(); }
00046
00047 std::string const &GetAbbreviation(uint16_t num) const;
00048
00049 std::string const &GetName(uint16_t num) const;
00050
00051 protected:
00052 void Add(std::string const &abbreviation, std::string const &name);
00053 void Insert(uint16_t num, std::string const &abbreviation, std::string const &name);
00054
00055 private:
00056 // Generated implementation, see gdcmDefaultGroupNames
00057 void FillDefaultGroupName();
00058
00059 GroupDict &operator=(const GroupDict &__val); // purposely not implemented
00060 GroupDict(const GroupDict &__val); // purposely not implemented
00061
00062 GroupStringVector Abbreviations;
00063 GroupStringVector Names;
00064 };
00065 //-----
00066 inline std::ostream& operator<<(std::ostream& __os, const GroupDict &__val)
00067 {
00068     size_t size = __val.Size();
00069     for(size_t i=0; i<size; ++i)
00070     {
00071         __os << std::hex << std::setw(4) << std::setfill( '0' ) << i << ", "
00072         << __val.GetAbbreviation((uint16_t)i) << ", " << __val.GetName((uint16_t)i) << "\n";
00073     }
00074     return __os;
00075 }
00076
00077 } // end namespace gdcm
00078
00079 #endif //GDCMGROUPDICT_H

```


13.105 gdcmSOPClassUIDToIOD.h File Reference

#include "gdcmUIDs.h"

Include dependency graph for gdcmSOPClassUIDToIOD.h:



Classes

- class [gdcm::SOPClassUIDToIOD](#)
Class convert a class SOP Class UID into [IOD](#).

Namespaces

- namespace [gdcm](#)

13.106 gdcmSOPClassUIDToIOD.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  /*
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMSOPCLASSUIDTOIOD_H
00016 #define GDCMSOPCLASSUIDTOIOD_H

```

```

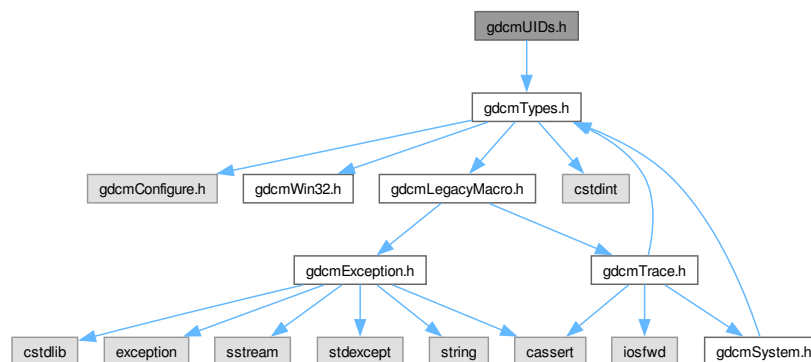
00017
00018 #include "gdcmUIDs.h"
00019
00020 namespace gdcm
00021 {
00022
00028 class GDCM_EXPORT SOPClassUIDToIOD
00029 {
00030 public:
00033 static const char *GetIOD(UIDs const & uid);
00034
00036 static unsigned int GetNumberOfSOPClassToIOD();
00037
00038 typedef const char* const (SOPClassUIDToIODType)[2];
00039 static SOPClassUIDToIODType* GetSOPClassUIDToIODs();
00040
00041 static SOPClassUIDToIODType& GetSOPClassUIDToIOD(unsigned int i);
00042
00043 static const char *GetSOPClassUIDFromIOD(const char *iod);
00044 static const char *GetIODFromSOPClassUID(const char *sopclassuid);
00045 };
00046
00047 } // end namespace gdcm
00048
00049 #endif //GDCMSOPCLASSUIDTOIOD_H

```

13.107 gdcmUIDs.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmUIDs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::UIDs`
all known uids

Namespaces

- namespace `gdcml`

Functions

- `std::ostream & gdcml::operator<<` (`std::ostream &_os`, `const UIDs &uid`)

13.108 gdcmlUIDs.h

[Go to the documentation of this file.](#)

```

00001
00002 // GENERATED FILE DO NOT EDIT
00003 // $ xsltproc UIDToC++.xsl Part6.xml > gdcmlUIDs.h
00004
00005
00006 /*=====
00007 Program: GDCM (Grassroots DICOM). A DICOM library
00008
00009 Copyright (c) 2006-2011 Mathieu Malaterre
00010 All rights reserved.
00011 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00012
00013 This software is distributed WITHOUT ANY WARRANTY; without even
00014 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00015 PURPOSE. See the above copyright notice for more information.
00016
00017
00018 =====*/
00019 #ifndef GDCMLUIDS_H
00020 #define GDCMLUIDS_H
00021
00022 #include "gdcmlTypes.h"
00023
00024 namespace gdcml
00025 {
00026
00027 class GDCML_EXPORT UIDs
00028 {
00029 public:
00030     typedef enum {
00031         uid_1_2_840_10008_1_1 = 1, // Verification SOP Class
00032         uid_1_2_840_10008_1_2 = 2, // Implicit VR Little Endian: Default Transfer Syntax for DICOM
00033         uid_1_2_840_10008_1_2_1 = 3, // Explicit VR Little Endian
00034         uid_1_2_840_10008_1_2_1_99 = 4, // Deflated Explicit VR Little Endian
00035         uid_1_2_840_10008_1_2_2 = 5, // Explicit VR Big Endian
00036         uid_1_2_840_10008_1_2_4_50 = 6, // JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image
00037         uid_1_2_840_10008_1_2_4_51 = 7, // JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit
00038         uid_1_2_840_10008_1_2_4_52 = 8, // JPEG Extended (Process 3 & 5)
00039         uid_1_2_840_10008_1_2_4_53 = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
00040         uid_1_2_840_10008_1_2_4_54 = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
00041         uid_1_2_840_10008_1_2_4_55 = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
00042         uid_1_2_840_10008_1_2_4_56 = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
00043         uid_1_2_840_10008_1_2_4_57 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
00044         uid_1_2_840_10008_1_2_4_58 = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
00045         uid_1_2_840_10008_1_2_4_59 = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
00046         uid_1_2_840_10008_1_2_4_60 = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
00047         uid_1_2_840_10008_1_2_4_61 = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
00048         uid_1_2_840_10008_1_2_4_62 = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
00049         uid_1_2_840_10008_1_2_4_63 = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
00050         uid_1_2_840_10008_1_2_4_64 = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
00051         uid_1_2_840_10008_1_2_4_65 = 21, // JPEG Lossless, Hierarchical (Process 28)
00052         uid_1_2_840_10008_1_2_4_66 = 22, // JPEG Lossless, Hierarchical (Process 29)
00053         uid_1_2_840_10008_1_2_4_70 = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection
00054         uid_1_2_840_10008_1_2_4_80 = 24, // JPEG-LS Lossless Image Compression
00055     }
00056
00057     //=====
00058
00059     //=====
00060
00061     //=====
00062
00063     //=====
00064
00065     //=====
00066
00067     //=====
00068
00069     //=====
00070
00071     //=====
00072
00073     //=====
00074
00075     //=====
00076
00077     //=====
00078
00079     //=====
00080
00081     //=====
00082
00083     //=====
00084
00085     //=====
00086
00087     //=====
00088
00089     //=====
00090
00091     //=====
00092
00093     //=====
00094
00095     //=====
00096
00097     //=====
00098
00099     //=====
00100
00101     //=====
00102
00103     //=====
00104
00105     //=====
00106
00107     //=====
00108
00109     //=====
00110
00111     //=====
00112
00113     //=====
00114
00115     //=====
00116
00117     //=====
00118
00119     //=====
00120
00121     //=====
00122
00123     //=====
00124
00125     //=====
00126
00127     //=====
00128
00129     //=====
00130
00131     //=====
00132
00133     //=====
00134
00135     //=====
00136
00137     //=====
00138
00139     //=====
00140
00141     //=====
00142
00143     //=====
00144
00145     //=====
00146
00147     //=====
00148
00149     //=====
00150
00151     //=====
00152
00153     //=====
00154
00155     //=====
00156
00157     //=====
00158
00159     //=====
00160
00161     //=====
00162
00163     //=====
00164
00165     //=====
00166
00167     //=====
00168
00169     //=====
00170
00171     //=====
00172
00173     //=====
00174
00175     //=====
00176
00177     //=====
00178
00179     //=====
00180
00181     //=====
00182
00183     //=====
00184
00185     //=====
00186
00187     //=====
00188
00189     //=====
00190
00191     //=====
00192
00193     //=====
00194
00195     //=====
00196
00197     //=====
00198
00199     //=====
00200
00201     //=====
00202
00203     //=====
00204
00205     //=====
00206
00207     //=====
00208
00209     //=====
00210
00211     //=====
00212
00213     //=====
00214
00215     //=====
00216
00217     //=====
00218
00219     //=====
00220
00221     //=====
00222
00223     //=====
00224
00225     //=====
00226
00227     //=====
00228
00229     //=====
00230
00231     //=====
00232
00233     //=====
00234
00235     //=====
00236
00237     //=====
00238
00239     //=====
00240
00241     //=====
00242
00243     //=====
00244
00245     //=====
00246
00247     //=====
00248
00249     //=====
00250
00251     //=====
00252
00253     //=====
00254
00255     //=====
00256
00257     //=====
00258
00259     //=====
00260
00261     //=====
00262
00263     //=====
00264
00265     //=====
00266
00267     //=====
00268
00269     //=====
00270
00271     //=====
00272
00273     //=====
00274
00275     //=====
00276
00277     //=====
00278
00279     //=====
00280
00281     //=====
00282
00283     //=====
00284
00285     //=====
00286
00287     //=====
00288
00289     //=====
00290
00291     //=====
00292
00293     //=====
00294
00295     //=====
00296
00297     //=====
00298
00299     //=====
00300
00301     //=====
00302
00303     //=====
00304
00305     //=====
00306
00307     //=====
00308
00309     //=====
00310
00311     //=====
00312
00313     //=====
00314
00315     //=====
00316
00317     //=====
00318
00319     //=====
00320
00321     //=====
00322
00323     //=====
00324
00325     //=====
00326
00327     //=====
00328
00329     //=====
00330
00331     //=====
00332
00333     //=====
00334
00335     //=====
00336
00337     //=====
00338
00339     //=====
00340
00341     //=====
00342
00343     //=====
00344
00345     //=====
00346
00347     //=====
00348
00349     //=====
00350
00351     //=====
00352
00353     //=====
00354
00355     //=====
00356
00357     //=====
00358
00359     //=====
00360
00361     //=====
00362
00363     //=====
00364
00365     //=====
00366
00367     //=====
00368
00369     //=====
00370
00371     //=====
00372
00373     //=====
00374
00375     //=====
00376
00377     //=====
00378
00379     //=====
00380
00381     //=====
00382
00383     //=====
00384
00385     //=====
00386
00387     //=====
00388
00389     //=====
00390
00391     //=====
00392
00393     //=====
00394
00395     //=====
00396
00397     //=====
00398
00399     //=====
00400
00401     //=====
00402
00403     //=====
00404
00405     //=====
00406
00407     //=====
00408
00409     //=====
00410
00411     //=====
00412
00413     //=====
00414
00415     //=====
00416
00417     //=====
00418
00419     //=====
00420
00421     //=====
00422
00423     //=====
00424
00425     //=====
00426
00427     //=====
00428
00429     //=====
00430
00431     //=====
00432
00433     //=====
00434
00435     //=====
00436
00437     //=====
00438
00439     //=====
00440
00441     //=====
00442
00443     //=====
00444
00445     //=====
00446
00447     //=====
00448
00449     //=====
00450
00451     //=====
00452
00453     //=====
00454
00455     //=====
00456
00457     //=====
00458
00459     //=====
00460
00461     //=====
00462
00463     //=====
00464
00465     //=====
00466
00467     //=====
00468
00469     //=====
00470
00471     //=====
00472
00473     //=====
00474
00475     //=====
00476
00477     //=====
00478
00479     //=====
00480
00481     //=====
00482
00483     //=====
00484
00485     //=====
00486
00487     //=====
00488
00489     //=====
00490
00491     //=====
00492
00493     //=====
00494
00495     //=====
00496
00497     //=====
00498
00499     //=====
00500
00501     //=====
00502
00503     //=====
00504
00505     //=====
00506
00507     //=====
00508
00509     //=====
00510
00511     //=====
00512
00513     //=====
00514
00515     //=====
00516
00517     //=====
00518
00519     //=====
00520
00521     //=====
00522
00523     //=====
00524
00525     //=====
00526
00527     //=====
00528
00529     //=====
00530
00531     //=====
00532
00533     //=====
00534
00535     //=====
00536
00537     //=====
00538
00539     //=====
00540
00541     //=====
00542
00543     //=====
00544
00545     //=====
00546
00547     //=====
00548
00549     //=====
00550
00551     //=====
00552
00553     //=====
00554
00555     //=====
00556
00557     //=====
00558
00559     //=====
00560
00561     //=====
00562
00563     //=====
00564
00565     //=====
00566
00567     //=====
00568
00569     //=====
00570
00571     //=====
00572
00573     //=====
00574
00575     //=====
00576
00577     //=====
00578
00579     //=====
00580
00581     //=====
00582
00583     //=====
00584
00585     //=====
00586
00587     //=====
00588
00589     //=====
00590
00591     //=====
00592
00593     //=====
00594
00595     //=====
00596
00597     //=====
00598
00599     //=====
00600
00601     //=====
00602
00603     //=====
00604
00605     //=====
00606
00607     //=====
00608
00609     //=====
00610
00611     //=====
00612
00613     //=====
00614
00615     //=====
00616
00617     //=====
00618
00619     //=====
00620
00621     //=====
00622
00623     //=====
00624
00625     //=====
00626
00627     //=====
00628
00629     //=====
00630
00631     //=====
00632
00633     //=====
00634
00635     //=====
00636
00637     //=====
00638
00639     //=====
00640
00641     //=====
00642
00643     //=====
00644
00645     //=====
00646
00647     //=====
00648
00649     //=====
00650
00651     //=====
00652
00653     //=====
00654
00655     //=====
00656
00657     //=====
00658
00659     //=====
00660
00661     //=====
00662
00663     //=====
00664
00665     //=====
00666
00667     //=====
00668
00669     //=====
00670
00671     //=====
00672
00673     //=====
00674
00675     //=====
00676
00677     //=====
00678
00679     //=====
00680
00681     //=====
00682
00683     //=====
00684
00685     //=====
00686
00687     //=====
00688
00689     //=====
00690
00691     //=====
00692
00693     //=====
00694
00695     //=====
00696
00697     //=====
00698
00699     //=====
00700
00701     //=====
00702
00703     //=====
00704
00705     //=====
00706
00707     //=====
00708
00709     //=====
00710
00711     //=====
00712
00713     //=====
00714
00715     //=====
00716
00717     //=====
00718
00719     //=====
00720
00721     //=====
00722
00723     //=====
00724
00725     //=====
00726
00727     //=====
00728
00729     //=====
00730
00731     //=====
00732
00733     //=====
00734
00735     //=====
00736
00737     //=====
00738
00739     //=====
00740
00741     //=====
00742
00743     //=====
00744
00745     //=====
00746
00747     //=====
00748
00749     //=====
00750
00751     //=====
00752
00753     //=====
00754
00755     //=====
00756
00757     //=====
00758
00759     //=====
00760
00761     //=====
00762
00763     //=====
00764
00765     //=====
00766
00767     //=====
00768
00769     //=====
00770
00771     //=====
00772
00773     //=====
00774
00775     //=====
00776
00777     //=====
00778
00779     //=====
00780
00781     //=====
00782
00783     //=====
00784
00785     //=====
00786
00787     //=====
00788
00789     //=====
00790
00791     //=====
00792
00793     //=====
00794
00795     //=====
00796
00797     //=====
00798
00799     //=====
00800
00801     //=====
00802
00803     //=====
00804
00805     //=====
00806
00807     //=====
00808
00809     //=====
00810
00811     //=====
00812
00813     //=====
00814
00815     //=====
00816
00817     //=====
00818
00819     //=====
00820
00821     //=====
00822
00823     //=====
00824
00825     //=====
00826
00827     //=====
00828
00829     //=====
00830
00831     //=====
00832
00833     //=====
00834
00835     //=====
00836
00837     //=====
00838
00839     //=====
00840
00841     //=====
00842
00843     //=====
00844
00845     //=====
00846
00847     //=====
00848
00849     //=====
00850
00851     //=====
00852
00853     //=====
00854
00855     //=====
00856
00857     //=====
00858
00859     //=====
00860
00861     //=====
00862
00863     //=====
00864
00865     //=====
00866
00867     //=====
00868
00869     //=====
00870
00871     //=====
00872
00873     //=====
00874
00875     //=====
00876
00877     //=====
00878
00879     //=====
00880
00881     //=====
00882
00883     //=====
00884
00885     //=====
00886
00887     //=====
00888
00889     //=====
00890
00891     //=====
00892
00893     //=====
00894
00895     //=====
00896
00897     //=====
00898
00899     //=====
00900
00901     //=====
00902
00903     //=====
00904
00905     //=====
00906
00907     //=====
00908
00909     //=====
00910
00911     //=====
00912
00913     //=====
00914
00915     //=====
00916
00917     //=====
00918
00919     //=====
00920
00921     //=====
00922
00923     //=====
00924
00925     //=====
00926
00927     //=====
00928
00929     //=====
00930
00931     //=====
00932
00933     //=====
00934
00935     //=====
00936
00937     //=====
00938
00939     //=====
00940
00941     //=====
00942
00943     //=====
00944
00945     //=====
00946
00947     //=====
00948
00949     //=====
00950
00951     //=====
00952
00953     //=====
00954
00955     //=====
00956
00957     //=====
00958
00959     //=====
00960
00961     //=====
00962
00963     //=====
00964
00965     //=====
00966
00967     //=====
00968
00969     //=====
00970
00971     //=====
00972
00973     //=====
00974
00975     //=====
00976
00977     //=====
00978
00979     //=====
00980
00981     //=====
00982
00983     //=====
00984
00985     //=====
00986
00987     //=====
00988
00989     //=====
00990
00991     //=====
00992
00993     //=====
00994
00995     //=====
00996
00997     //=====
00998
00999     //=====
01000
01001     //=====
01002
01003     //=====
01004
01005     //=====
01006
01007     //=====
01008
01009     //=====
01010
01011     //=====
01012
01013     //=====
01014
01015     //=====
01016
01017     //=====
01018
01019     //=====
01020
01021     //=====
01022
01023     //=====
01024
01025     //=====
01026
01027     //=====
01028
01029     //=====
01030
01031     //=====
01032
01033     //=====
01034
01035     //=====
01036
01037     //=====
01038
01039     //=====
01040
01041     //=====
01042
01043     //=====
01044
01045     //=====
01046
01047     //=====
01048
01049     //=====
01050
01051     //=====
01052
01053     //=====
01054
01055     //=====
01056
01057     //=====
01058
01059     //=====
01060
01061     //=====
01062
01063     //=====
01064
01065     //=====
01066
01067     //=====
01068
01069     //=====
01070
01071     //=====
01072
01073     //=====
01074
01075     //=====
01076
01077     //=====
01078
01079     //=====
01080
01081     //=====
01082
01083     //=====
01084
01085     //=====
01086
01087     //=====
01088
01089     //=====
01090
01091     //=====
01092
01093     //=====
01094
01095     //=====
01096
01097     //=====
01098
01099     //=====
01100
01101     //=====
01102
01103     //=====
01104
01105     //=====
01106
01107     //=====
01108
01109     //=====
01110
01111     //=====
01112
01113     //=====
01114
01115     //=====
01116
01117     //=====
01118
01119     //=====
01120
01121     //=====
01122
01123     //=====
01124
01125     //=====
01126
01127     //=====
01128
01129     //=====
01130
01131     //=====
01132
01133     //=====
01134
01135     //=====
01136
01137     //=====
01138
01139     //=====
01140
01141     //=====
01142
01143     //=====
01144
01145     //=====
01146
01147     //=====
01148
01149     //=====
01150
01151     //=====
01152
01153     //=====
01154
01155     //=====
01156
01157     //=====
01158
01159     //=====
01160
01161     //=====
01162
01163     //=====
01164
01165     //=====
01166
01167     //=====
01168
01169     //=====
01170
01171     //=====
01172
01173     //=====
01174
01175     //=====
01176
01177     //=====
01178
01179     //=====
01180
01181     //=====
01182
01183     //=====
01184
01185     //=====
01186
01187     //=====
01188
01189     //=====
01190
01191     //=====
01192
01193     //=====
01194
01195     //=====
01196
01197     //=====
01198
01199     //=====
01200
01201     //=====
01202
01203     //=====
01204
01205     //=====
01206
01207     //=====
01208
01209     //=====
01210
01211     //=====
01212
01213     //=====
01214
01215     //=====
01216
01217     //=====
01218
01219     //=====
01220
01221     //=====
01222
01223     //=====
01224
01225     //=====
01226
01227     //=====
01228
01229     //=====
01230
01231     //=====
01232
01233     //=====
01234
01235     //=====
01236
01237     //=====
01238
01239     //=====
01240
01241     //=====
01242
01243     //=====
01244
01245     //=====
01246
01247     //=====
01248
01249     //=====
01250
01251     //=====
01252
01253     //=====
01254
01255     //=====
01256
01257     //=====
01258
01259     //=====
01260
01261     //=====
01262
01263     //=====
01264
01265     //=====
01266
01267     //=====
01268
01269     //=====
01270
01271     //=====
01272
01273     //=====
01274
01275     //=====
01276
01277     //=====
01278
01279     //=====
01280
01281     //=====
01282
01283     //=====
01284
01285     //=====
01286
01287     //=====
01288
01289     //=====
01290
01291     //=====
01292
01293     //=====
01294
01295     //=====
01296
01297     //=====
01298
01299     //=====
01300
01301     //=====
01302
01303     //=====
01304
01305     //=====
01306
01307     //=====
01308
01309     //=====
01310
01311     //=====
01312
01313     //=====
01314
01315     //=====
01316
01317     //=====
01318
01319     //=====
01320
01321     //=====
01322
01323     //=====
01324
01325     //=====
01326
01327     //=====
01328
01329     //=====
01330
01331     //=====
01332
01333     //=====
01334
01335     //=====
01336
01337     //=====
01338
01339     //=====
01340
01341     //=====
01342
01343     //=====
01344
01345     //=====
01346
01347     //=====
01348
01349     //=====
01350
01351     //=====
01352
01353     //=====
01354
01355     //=====
01356
01357     //=====
01358
01359     //=====
01360
01361     //=====
01362
01363     //=====
01364
01365     //=====
01366
01367     //=====
01368
01369     //=====
01370
01371     //=====
01372
01373     //=====
01374
01375     //=====
01376
01377     //=====
01378
01379     //=====
01380
01381     //=====
01382
01383     //=====
01384
01385     //=====
01386
01387     //=====
01388
01389     //=====
01390
01391     //=====
01392
01393     //=====
01394
01395     //=====
01396
01397     //=====
01398
01399     //=====
01400
01401     //=====
01402
01403     //=====
01404
01405     //=====
01406
01407     //=====
01408
01409     //=====
01410
01411     //=====
01412
01413     //=====
01414
01415     //=====
01416
01417     //=====
01418
01419     //=====
01420
01421     //=====
01422
01423     //=====
01424
01425     //=====
01426
01427     //=====
01428
01429     //=====
01430
01431     //=====
01432
01433     //=====
01434
01435     //=====
01436
01437     //=====
01438
01439     //=====
01440
01441     //=====
01442
01443
```

```

00058 uid_1_2_840_10008_1_2_4_81 = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
00059 uid_1_2_840_10008_1_2_4_90 = 26, // JPEG 2000 Image Compression (Lossless Only)
00060 uid_1_2_840_10008_1_2_4_91 = 27, // JPEG 2000 Image Compression
00061 uid_1_2_840_10008_1_2_4_92 = 28, // JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)
00062 uid_1_2_840_10008_1_2_4_93 = 29, // JPEG 2000 Part 2 Multi-component Image Compression
00063 uid_1_2_840_10008_1_2_4_94 = 30, // JPIP Referenced
00064 uid_1_2_840_10008_1_2_4_95 = 31, // JPIP Referenced Deflate
00065 uid_1_2_840_10008_1_2_4_100 = 32, // MPEG2 Main Profile @ Main Level
00066 uid_1_2_840_10008_1_2_5 = 33, // RLE Lossless
00067 uid_1_2_840_10008_1_2_6_1 = 34, // RFC 2557 MIME encapsulation
00068 uid_1_2_840_10008_1_2_6_2 = 35, // XML Encoding
00069 uid_1_2_840_10008_1_3_10 = 36, // Media Storage Directory Storage
00070 uid_1_2_840_10008_1_4_1_1 = 37, // Talairach Brain Atlas Frame of Reference
00071 uid_1_2_840_10008_1_4_1_2 = 38, // SPM2 T1 Frame of Reference
00072 uid_1_2_840_10008_1_4_1_3 = 39, // SPM2 T2 Frame of Reference
00073 uid_1_2_840_10008_1_4_1_4 = 40, // SPM2 PD Frame of Reference
00074 uid_1_2_840_10008_1_4_1_5 = 41, // SPM2 EPI Frame of Reference
00075 uid_1_2_840_10008_1_4_1_6 = 42, // SPM2 FIL T1 Frame of Reference
00076 uid_1_2_840_10008_1_4_1_7 = 43, // SPM2 PET Frame of Reference
00077 uid_1_2_840_10008_1_4_1_8 = 44, // SPM2 TRANSM Frame of Reference
00078 uid_1_2_840_10008_1_4_1_9 = 45, // SPM2 SPECT Frame of Reference
00079 uid_1_2_840_10008_1_4_1_10 = 46, // SPM2 GRAY Frame of Reference
00080 uid_1_2_840_10008_1_4_1_11 = 47, // SPM2 WHITE Frame of Reference
00081 uid_1_2_840_10008_1_4_1_12 = 48, // SPM2 CSF Frame of Reference
00082 uid_1_2_840_10008_1_4_1_13 = 49, // SPM2 BRAINMASK Frame of Reference
00083 uid_1_2_840_10008_1_4_1_14 = 50, // SPM2 AVG305T1 Frame of Reference
00084 uid_1_2_840_10008_1_4_1_15 = 51, // SPM2 AVG152T1 Frame of Reference
00085 uid_1_2_840_10008_1_4_1_16 = 52, // SPM2 AVG152T2 Frame of Reference
00086 uid_1_2_840_10008_1_4_1_17 = 53, // SPM2 AVG152PD Frame of Reference
00087 uid_1_2_840_10008_1_4_1_18 = 54, // SPM2 SINGLESUBJT1 Frame of Reference
00088 uid_1_2_840_10008_1_4_2_1 = 55, // ICBM 452 T1 Frame of Reference
00089 uid_1_2_840_10008_1_4_2_2 = 56, // ICBM Single Subject MRI Frame of Reference
00090 uid_1_2_840_10008_1_9 = 57, // Basic Study Content Notification SOP Class
00091 uid_1_2_840_10008_1_20_1 = 58, // Storage Commitment Push Model SOP Class
00092 uid_1_2_840_10008_1_20_1_1 = 59, // Storage Commitment Push Model SOP Instance
00093 uid_1_2_840_10008_1_20_2 = 60, // Storage Commitment Pull Model SOP Class
00094 uid_1_2_840_10008_1_20_2_1 = 61, // Storage Commitment Pull Model SOP Instance
00095 uid_1_2_840_10008_1_40 = 62, // Procedural Event Logging SOP Class
00096 uid_1_2_840_10008_1_40_1 = 63, // Procedural Event Logging SOP Instance
00097 uid_1_2_840_10008_1_42 = 64, // Substance Administration Logging SOP Class
00098 uid_1_2_840_10008_1_42_1 = 65, // Substance Administration Logging SOP Instance
00099 uid_1_2_840_10008_2_6_1 = 66, // DICOM UID Registry
00100 uid_1_2_840_10008_2_16_4 = 67, // DICOM Controlled Terminology
00101 uid_1_2_840_10008_3_1_1_1 = 68, // DICOM Application Context Name
00102 uid_1_2_840_10008_3_1_2_1_1 = 69, // Detached Patient Management SOP Class
00103 uid_1_2_840_10008_3_1_2_1_4 = 70, // Detached Patient Management Meta SOP Class
00104 uid_1_2_840_10008_3_1_2_2_1 = 71, // Detached Visit Management SOP Class
00105 uid_1_2_840_10008_3_1_2_3_1 = 72, // Detached Study Management SOP Class
00106 uid_1_2_840_10008_3_1_2_3_2 = 73, // Study Component Management SOP Class
00107 uid_1_2_840_10008_3_1_2_3_3 = 74, // Modality Performed Procedure Step SOP Class
00108 uid_1_2_840_10008_3_1_2_3_4 = 75, // Modality Performed Procedure Step Retrieve SOP Class
00109 uid_1_2_840_10008_3_1_2_3_5 = 76, // Modality Performed Procedure Step Notification SOP Class
00110 uid_1_2_840_10008_3_1_2_5_1 = 77, // Detached Results Management SOP Class
00111 uid_1_2_840_10008_3_1_2_5_4 = 78, // Detached Results Management Meta SOP Class
00112 uid_1_2_840_10008_3_1_2_5_5 = 79, // Detached Study Management Meta SOP Class
00113 uid_1_2_840_10008_3_1_2_6_1 = 80, // Detached Interpretation Management SOP Class
00114 uid_1_2_840_10008_4_2 = 81, // Storage Service Class
00115 uid_1_2_840_10008_5_1_1_1 = 82, // Basic Film Session SOP Class
00116 uid_1_2_840_10008_5_1_1_2 = 83, // Basic Film Box SOP Class
00117 uid_1_2_840_10008_5_1_1_4 = 84, // Basic Grayscale Image Box SOP Class
00118 uid_1_2_840_10008_5_1_1_4_1 = 85, // Basic Color Image Box SOP Class
00119 uid_1_2_840_10008_5_1_1_4_2 = 86, // Referenced Image Box SOP Class
00120 uid_1_2_840_10008_5_1_1_9 = 87, // Basic Grayscale Print Management Meta SOP Class
00121 uid_1_2_840_10008_5_1_1_9_1 = 88, // Referenced Grayscale Print Management Meta SOP Class
00122 uid_1_2_840_10008_5_1_1_14 = 89, // Print Job SOP Class
00123 uid_1_2_840_10008_5_1_1_15 = 90, // Basic Annotation Box SOP Class
00124 uid_1_2_840_10008_5_1_1_16 = 91, // Printer SOP Class
00125 uid_1_2_840_10008_5_1_1_16_376 = 92, // Printer Configuration Retrieval SOP Class
00126 uid_1_2_840_10008_5_1_1_17 = 93, // Printer SOP Instance
00127 uid_1_2_840_10008_5_1_1_17_376 = 94, // Printer Configuration Retrieval SOP Instance
00128 uid_1_2_840_10008_5_1_1_18 = 95, // Basic Color Print Management Meta SOP Class
00129 uid_1_2_840_10008_5_1_1_18_1 = 96, // Referenced Color Print Management Meta SOP Class
00130 uid_1_2_840_10008_5_1_1_22 = 97, // VOI LUT Box SOP Class
00131 uid_1_2_840_10008_5_1_1_23 = 98, // Presentation LUT SOP Class
00132 uid_1_2_840_10008_5_1_1_24 = 99, // Image Overlay Box SOP Class
00133 uid_1_2_840_10008_5_1_1_24_1 = 100, // Basic Print Image Overlay Box SOP Class
00134 uid_1_2_840_10008_5_1_1_25 = 101, // Print Queue SOP Instance
00135 uid_1_2_840_10008_5_1_1_26 = 102, // Print Queue Management SOP Class
00136 uid_1_2_840_10008_5_1_1_27 = 103, // Stored Print Storage SOP Class
00137 uid_1_2_840_10008_5_1_1_29 = 104, // Hardcopy Grayscale Image Storage SOP Class
00138 uid_1_2_840_10008_5_1_1_30 = 105, // Hardcopy Color Image Storage SOP Class

```

00139 uid_1_2_840_10008_5_1_1_31 = 106, // Pull Print Request SOP Class
00140 uid_1_2_840_10008_5_1_1_32 = 107, // Pull Stored Print Management Meta SOP Class
00141 uid_1_2_840_10008_5_1_1_33 = 108, // Media Creation Management SOP Class UID
00142 uid_1_2_840_10008_5_1_4_1_1_1 = 109, // Computed Radiography Image Storage
00143 uid_1_2_840_10008_5_1_4_1_1_1_1 = 110, // Digital X-Ray Image Storage - For Presentation
00144 uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111, // Digital X-Ray Image Storage - For Processing
00145 uid_1_2_840_10008_5_1_4_1_1_1_1_2 = 112, // Digital Mammography X-Ray Image Storage - For Presentation
00146 uid_1_2_840_10008_5_1_4_1_1_1_1_2_1 = 113, // Digital Mammography X-Ray Image Storage - For Processing
00147 uid_1_2_840_10008_5_1_4_1_1_1_1_3 = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
00148 uid_1_2_840_10008_5_1_4_1_1_1_1_3_1 = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
00149 uid_1_2_840_10008_5_1_4_1_1_2 = 116, // CT Image Storage
00150 uid_1_2_840_10008_5_1_4_1_1_2_1 = 117, // Enhanced CT Image Storage
00151 uid_1_2_840_10008_5_1_4_1_1_3 = 118, // Ultrasound Multi-frame Image Storage
00152 uid_1_2_840_10008_5_1_4_1_1_3_1 = 119, // Ultrasound Multi-frame Image Storage
00153 uid_1_2_840_10008_5_1_4_1_1_4 = 120, // MR Image Storage
00154 uid_1_2_840_10008_5_1_4_1_1_4_1 = 121, // Enhanced MR Image Storage
00155 uid_1_2_840_10008_5_1_4_1_1_4_2 = 122, // MR Spectroscopy Storage
00156 uid_1_2_840_10008_5_1_4_1_1_5 = 123, // Nuclear Medicine Image Storage
00157 uid_1_2_840_10008_5_1_4_1_1_6 = 124, // Ultrasound Image Storage
00158 uid_1_2_840_10008_5_1_4_1_1_6_1 = 125, // Ultrasound Image Storage
00159 uid_1_2_840_10008_5_1_4_1_1_7 = 126, // Secondary Capture Image Storage
00160 uid_1_2_840_10008_5_1_4_1_1_7_1 = 127, // Multi-frame Single Bit Secondary Capture Image Storage
00161 uid_1_2_840_10008_5_1_4_1_1_7_2 = 128, // Multi-frame Grayscale Byte Secondary Capture Image Storage
00162 uid_1_2_840_10008_5_1_4_1_1_7_3 = 129, // Multi-frame Grayscale Word Secondary Capture Image Storage
00163 uid_1_2_840_10008_5_1_4_1_1_7_4 = 130, // Multi-frame True Color Secondary Capture Image Storage
00164 uid_1_2_840_10008_5_1_4_1_1_8 = 131, // Standalone Overlay Storage
00165 uid_1_2_840_10008_5_1_4_1_1_9 = 132, // Standalone Curve Storage
00166 uid_1_2_840_10008_5_1_4_1_1_9_1 = 133, // Waveform Storage - Trial
00167 uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134, // 12-lead ECG Waveform Storage
00168 uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135, // General ECG Waveform Storage
00169 uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136, // Ambulatory ECG Waveform Storage
00170 uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137, // Hemodynamic Waveform Storage
00171 uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138, // Cardiac Electrophysiology Waveform Storage
00172 uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139, // Basic Voice Audio Waveform Storage
00173 uid_1_2_840_10008_5_1_4_1_1_10 = 140, // Standalone Modality LUT Storage
00174 uid_1_2_840_10008_5_1_4_1_1_11 = 141, // Standalone VOI LUT Storage
00175 uid_1_2_840_10008_5_1_4_1_1_11_1 = 142, // Grayscale Softcopy Presentation State Storage SOP Class
00176 uid_1_2_840_10008_5_1_4_1_1_11_2 = 143, // Color Softcopy Presentation State Storage SOP Class
00177 uid_1_2_840_10008_5_1_4_1_1_11_3 = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
00178 uid_1_2_840_10008_5_1_4_1_1_11_4 = 145, // Blending Softcopy Presentation State Storage SOP Class
00179 uid_1_2_840_10008_5_1_4_1_1_12_1 = 146, // X-Ray Angiographic Image Storage
00180 uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147, // Enhanced XA Image Storage
00181 uid_1_2_840_10008_5_1_4_1_1_12_2 = 148, // X-Ray Radiofluoroscopic Image Storage
00182 uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149, // Enhanced XRF Image Storage
00183 uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150, // X-Ray 3D Angiographic Image Storage
00184 uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151, // X-Ray 3D Craniofacial Image Storage
00185 uid_1_2_840_10008_5_1_4_1_1_12_3 = 152, // X-Ray Angiographic Bi-Plane Image Storage
00186 uid_1_2_840_10008_5_1_4_1_1_20 = 153, // Nuclear Medicine Image Storage
00187 uid_1_2_840_10008_5_1_4_1_1_66 = 154, // Raw Data Storage
00188 uid_1_2_840_10008_5_1_4_1_1_66_1 = 155, // Spatial Registration Storage
00189 uid_1_2_840_10008_5_1_4_1_1_66_2 = 156, // Spatial Fiducials Storage
00190 uid_1_2_840_10008_5_1_4_1_1_66_3 = 157, // Deformable Spatial Registration Storage
00191 uid_1_2_840_10008_5_1_4_1_1_66_4 = 158, // Segmentation Storage
00192 uid_1_2_840_10008_5_1_4_1_1_67 = 159, // Real World Value Mapping Storage
00193 uid_1_2_840_10008_5_1_4_1_1_77_1 = 160, // VL Image Storage - Trial
00194 uid_1_2_840_10008_5_1_4_1_1_77_2 = 161, // VL Multi-frame Image Storage - Trial
00195 uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162, // VL Endoscopic Image Storage
00196 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163, // Video Endoscopic Image Storage
00197 uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164, // VL Microscopic Image Storage
00198 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165, // Video Microscopic Image Storage
00199 uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166, // VL Slide-Coordinates Microscopic Image Storage
00200 uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167, // VL Photographic Image Storage
00201 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168, // Video Photographic Image Storage
00202 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169, // Ophthalmic Photography 8 Bit Image Storage
00203 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170, // Ophthalmic Photography 16 Bit Image Storage
00204 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171, // Stereometric Relationship Storage
00205 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172, // Ophthalmic Tomography Image Storage
00206 uid_1_2_840_10008_5_1_4_1_1_88_1 = 173, // Text SR Storage - Trial
00207 uid_1_2_840_10008_5_1_4_1_1_88_2 = 174, // Audio SR Storage - Trial
00208 uid_1_2_840_10008_5_1_4_1_1_88_3 = 175, // Detail SR Storage - Trial
00209 uid_1_2_840_10008_5_1_4_1_1_88_4 = 176, // Comprehensive SR Storage - Trial
00210 uid_1_2_840_10008_5_1_4_1_1_88_11 = 177, // Basic Text SR Storage
00211 uid_1_2_840_10008_5_1_4_1_1_88_22 = 178, // Enhanced SR Storage
00212 uid_1_2_840_10008_5_1_4_1_1_88_33 = 179, // Comprehensive SR Storage
00213 uid_1_2_840_10008_5_1_4_1_1_88_40 = 180, // Procedure Log Storage
00214 uid_1_2_840_10008_5_1_4_1_1_88_50 = 181, // Mammography CAD SR Storage
00215 uid_1_2_840_10008_5_1_4_1_1_88_59 = 182, // Key Object Selection Document Storage
00216 uid_1_2_840_10008_5_1_4_1_1_88_65 = 183, // Chest CAD SR Storage
00217 uid_1_2_840_10008_5_1_4_1_1_88_67 = 184, // X-Ray Radiation Dose SR Storage
00218 uid_1_2_840_10008_5_1_4_1_1_104_1 = 185, // Encapsulated PDF Storage
00219 uid_1_2_840_10008_5_1_4_1_1_104_2 = 186, // Encapsulated CDA Storage

```

00220 uid_1_2_840_10008_5_1_4_1_1_128 = 187, // Positron Emission Tomography Image Storage
00221 uid_1_2_840_10008_5_1_4_1_1_129 = 188, // Standalone PET Curve Storage
00222 uid_1_2_840_10008_5_1_4_1_1_481_1 = 189, // RT Image Storage
00223 uid_1_2_840_10008_5_1_4_1_1_481_2 = 190, // RT Dose Storage
00224 uid_1_2_840_10008_5_1_4_1_1_481_3 = 191, // RT Structure Set Storage
00225 uid_1_2_840_10008_5_1_4_1_1_481_4 = 192, // RT Beams Treatment Record Storage
00226 uid_1_2_840_10008_5_1_4_1_1_481_5 = 193, // RT Plan Storage
00227 uid_1_2_840_10008_5_1_4_1_1_481_6 = 194, // RT Brachy Treatment Record Storage
00228 uid_1_2_840_10008_5_1_4_1_1_481_7 = 195, // RT Treatment Summary Record Storage
00229 uid_1_2_840_10008_5_1_4_1_1_481_8 = 196, // RT Ion Plan Storage
00230 uid_1_2_840_10008_5_1_4_1_1_481_9 = 197, // RT Ion Beams Treatment Record Storage
00231 uid_1_2_840_10008_5_1_4_1_2_1_1 = 198, // Patient Root Query/Retrieve Information Model - FIND
00232 uid_1_2_840_10008_5_1_4_1_2_1_2 = 199, // Patient Root Query/Retrieve Information Model - MOVE
00233 uid_1_2_840_10008_5_1_4_1_2_1_3 = 200, // Patient Root Query/Retrieve Information Model - GET
00234 uid_1_2_840_10008_5_1_4_1_2_2_1 = 201, // Study Root Query/Retrieve Information Model - FIND
00235 uid_1_2_840_10008_5_1_4_1_2_2_2 = 202, // Study Root Query/Retrieve Information Model - MOVE
00236 uid_1_2_840_10008_5_1_4_1_2_2_3 = 203, // Study Root Query/Retrieve Information Model - GET
00237 uid_1_2_840_10008_5_1_4_1_2_3_1 = 204, // Patient/Study Only Query/Retrieve Information Model - FIND
00238 uid_1_2_840_10008_5_1_4_1_2_3_2 = 205, // Patient/Study Only Query/Retrieve Information Model - MOVE
00239 uid_1_2_840_10008_5_1_4_1_2_3_3 = 206, // Patient/Study Only Query/Retrieve Information Model - GET
00240 uid_1_2_840_10008_5_1_4_1_31 = 207, // Modality Worklist Information Model - FIND
00241 uid_1_2_840_10008_5_1_4_32_1 = 208, // General Purpose Worklist Information Model - FIND
00242 uid_1_2_840_10008_5_1_4_32_2 = 209, // General Purpose Scheduled Procedure Step SOP Class
00243 uid_1_2_840_10008_5_1_4_32_3 = 210, // General Purpose Performed Procedure Step SOP Class
00244 uid_1_2_840_10008_5_1_4_32 = 211, // General Purpose Worklist Management Meta SOP Class
00245 uid_1_2_840_10008_5_1_4_33 = 212, // Instance Availability Notification SOP Class
00246 uid_1_2_840_10008_5_1_4_34_1 = 213, // RT Beams Delivery Instruction Storage (Supplement 74 Frozen Draft)
00247 uid_1_2_840_10008_5_1_4_34_2 = 214, // RT Conventional Machine Verification (Supplement 74 Frozen Draft)
00248 uid_1_2_840_10008_5_1_4_34_3 = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
00249 uid_1_2_840_10008_5_1_4_34_4 = 216, // Unified Worklist and Procedure Step Service Class
00250 uid_1_2_840_10008_5_1_4_34_4_1 = 217, // Unified Procedure Step - Push SOP Class
00251 uid_1_2_840_10008_5_1_4_34_4_2 = 218, // Unified Procedure Step - Watch SOP Class
00252 uid_1_2_840_10008_5_1_4_34_4_3 = 219, // Unified Procedure Step - Pull SOP Class
00253 uid_1_2_840_10008_5_1_4_34_4_4 = 220, // Unified Procedure Step - Event SOP Class
00254 uid_1_2_840_10008_5_1_4_34_5 = 221, // Unified Worklist and Procedure Step SOP Instance
00255 uid_1_2_840_10008_5_1_4_37_1 = 222, // General Relevant Patient Information Query
00256 uid_1_2_840_10008_5_1_4_37_2 = 223, // Breast Imaging Relevant Patient Information Query
00257 uid_1_2_840_10008_5_1_4_37_3 = 224, // Cardiac Relevant Patient Information Query
00258 uid_1_2_840_10008_5_1_4_38_1 = 225, // Hanging Protocol Storage
00259 uid_1_2_840_10008_5_1_4_38_2 = 226, // Hanging Protocol Information Model - FIND
00260 uid_1_2_840_10008_5_1_4_38_3 = 227, // Hanging Protocol Information Model - MOVE
00261 uid_1_2_840_10008_5_1_4_41 = 228, // Product Characteristics Query SOP Class
00262 uid_1_2_840_10008_5_1_4_42 = 229, // Substance Approval Query SOP Class
00263 uid_1_2_840_10008_15_0_3_1 = 230, // dicomDeviceName
00264 uid_1_2_840_10008_15_0_3_2 = 231, // dicomDescription
00265 uid_1_2_840_10008_15_0_3_3 = 232, // dicomManufacturer
00266 uid_1_2_840_10008_15_0_3_4 = 233, // dicomManufacturerModelName
00267 uid_1_2_840_10008_15_0_3_5 = 234, // dicomSoftwareVersion
00268 uid_1_2_840_10008_15_0_3_6 = 235, // dicomVendorData
00269 uid_1_2_840_10008_15_0_3_7 = 236, // dicomAETitle
00270 uid_1_2_840_10008_15_0_3_8 = 237, // dicomNetworkConnectionReference
00271 uid_1_2_840_10008_15_0_3_9 = 238, // dicomApplicationCluster
00272 uid_1_2_840_10008_15_0_3_10 = 239, // dicomAssociationInitiator
00273 uid_1_2_840_10008_15_0_3_11 = 240, // dicomAssociationAcceptor
00274 uid_1_2_840_10008_15_0_3_12 = 241, // dicomHostname
00275 uid_1_2_840_10008_15_0_3_13 = 242, // dicomPort
00276 uid_1_2_840_10008_15_0_3_14 = 243, // dicomSOPClass
00277 uid_1_2_840_10008_15_0_3_15 = 244, // dicomTransferRole
00278 uid_1_2_840_10008_15_0_3_16 = 245, // dicomTransferSyntax
00279 uid_1_2_840_10008_15_0_3_17 = 246, // dicomPrimaryDeviceType
00280 uid_1_2_840_10008_15_0_3_18 = 247, // dicomRelatedDeviceReference
00281 uid_1_2_840_10008_15_0_3_19 = 248, // dicomPreferredCalledAETitle
00282 uid_1_2_840_10008_15_0_3_20 = 249, // dicomTLSCyphersuite
00283 uid_1_2_840_10008_15_0_3_21 = 250, // dicomAuthorizedNodeCertificateReference
00284 uid_1_2_840_10008_15_0_3_22 = 251, // dicomThisNodeCertificateReference
00285 uid_1_2_840_10008_15_0_3_23 = 252, // dicomInstalled
00286 uid_1_2_840_10008_15_0_3_24 = 253, // dicomStationName
00287 uid_1_2_840_10008_15_0_3_25 = 254, // dicomDeviceSerialNumber
00288 uid_1_2_840_10008_15_0_3_26 = 255, // dicomInstitutionName
00289 uid_1_2_840_10008_15_0_3_27 = 256, // dicomInstitutionAddress
00290 uid_1_2_840_10008_15_0_3_28 = 257, // dicomInstitutionDepartmentName
00291 uid_1_2_840_10008_15_0_3_29 = 258, // dicomIssuerOfPatientID
00292 uid_1_2_840_10008_15_0_3_30 = 259, // dicomPreferredCallingAETitle
00293 uid_1_2_840_10008_15_0_3_31 = 260, // dicomSupportedCharacterSet
00294 uid_1_2_840_10008_15_0_4_1 = 261, // dicomConfigurationRoot
00295 uid_1_2_840_10008_15_0_4_2 = 262, // dicomDevicesRoot
00296 uid_1_2_840_10008_15_0_4_3 = 263, // dicomUniqueAETitlesRegistryRoot
00297 uid_1_2_840_10008_15_0_4_4 = 264, // dicomDevice
00298 uid_1_2_840_10008_15_0_4_5 = 265, // dicomNetworkAE
00299 uid_1_2_840_10008_15_0_4_6 = 266, // dicomNetworkConnection
00300 uid_1_2_840_10008_15_0_4_7 = 267, // dicomUniqueAETitle

```



```

00301 uid_1_2_840_10008_15_0_4_8 = 268, // dicomTransferCapability
00302 //
00303 uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269, // VL Whole Slide Microscopy
00304 uid_1_2_840_10008_5_1_4_1_1_6_2 = 270, // Enhanced US Volume Storage
00305 uid_1_2_840_10008_5_1_4_1_1_66_5 = 271, // Surface Segmentation Storage
00306 uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272, // Breast Tomosynthesis Image Storage
00307 uid_1_2_840_10008_5_1_4_1_1_2_2 = 273, // Legacy Converted Enhanced CT
00308 uid_1_2_840_10008_5_1_4_1_1_4_4 = 274, // Legacy Converted Enhanced MR
00309 uid_1_2_840_10008_5_1_4_1_1_128_1 = 275, // Legacy Converted Enhanced PET
00310 uid_1_2_840_10008_1_2_4_101 = 276, // MPEG2 Main Profile High Level
00311 uid_1_2_840_10008_1_2_4_102 = 277, // MPEG-4 AVC/H.264 High Profile Lev. 4.1
00312 uid_1_2_840_10008_1_2_4_103 = 278, // MPEG-4 AVC/H.264 BD-comp High Profile Lev. 4.1
00313 //
00315 //
00316 // 2019b
00317 //
00318 uid_1_2_840_10008_1_5_2 = 279,
00319 uid_1_2_840_10008_1_5_3 = 280,
00320 uid_1_2_840_10008_1_5_4 = 281,
00321 uid_1_2_840_10008_1_5_5 = 282,
00322 uid_1_2_840_10008_1_5_6 = 283,
00323 uid_1_2_840_10008_1_5_7 = 284,
00324 uid_1_2_840_10008_1_5_8 = 285,
00325 uid_1_2_840_10008_1_20 = 286,
00326 uid_1_2_840_10008_2_16_5 = 287,
00327 uid_1_2_840_10008_2_16_6 = 288,
00328 uid_1_2_840_10008_2_16_7 = 289,
00329 uid_1_2_840_10008_2_16_8 = 290,
00330 uid_1_2_840_10008_2_16_9 = 291,
00331 uid_1_2_840_10008_2_16_10 = 292,
00332 uid_1_2_840_10008_2_16_11 = 293,
00333 uid_1_2_840_10008_2_16_12 = 294,
00334 uid_1_2_840_10008_2_16_13 = 295,
00335 uid_1_2_840_10008_2_16_14 = 296,
00336 uid_1_2_840_10008_5_1_1_40 = 297,
00337 uid_1_2_840_10008_5_1_1_40_1 = 298,
00338 uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299,
00339 uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300,
00340 uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301,
00341 uid_1_2_840_10008_5_1_4_1_1_11_5 = 302,
00342 uid_1_2_840_10008_5_1_4_1_1_11_6 = 303,
00343 uid_1_2_840_10008_1_2_4_104 = 304,
00344 uid_1_2_840_10008_1_2_4_105 = 305,
00345 uid_1_2_840_10008_1_2_4_106 = 306,
00346 uid_1_2_840_10008_1_2_4_107 = 307,
00347 uid_1_2_840_10008_1_2_4_108 = 308,
00348 uid_1_2_840_10008_1_5_1 = 309,
00349 uid_1_2_840_10008_5_1_4_1_1_11_7 = 310,
00350 uid_1_2_840_10008_5_1_4_1_1_11_8 = 311,
00351 uid_1_2_840_10008_5_1_4_1_1_11_9 = 312,
00352 uid_1_2_840_10008_5_1_4_1_1_11_10 = 313,
00353 uid_1_2_840_10008_5_1_4_1_1_11_11 = 314,
00354 uid_1_2_840_10008_5_1_4_1_1_12_77 = 315,
00355 uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316,
00356 uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317,
00357 uid_1_2_840_10008_5_1_4_1_1_14_1 = 318,
00358 uid_1_2_840_10008_5_1_4_1_1_14_2 = 319,
00359 uid_1_2_840_10008_5_1_4_1_1_30 = 320,
00360 uid_1_2_840_10008_5_1_4_1_1_40 = 321,
00361 uid_1_2_840_10008_5_1_4_1_1_66_6 = 322,
00362 uid_1_2_840_10008_5_1_4_1_1_68_1 = 323,
00363 uid_1_2_840_10008_5_1_4_1_1_68_2 = 324,
00364 uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325,
00365 uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326,
00366 uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327,
00367 uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328,
00368 uid_1_2_840_10008_5_1_4_1_1_78_1 = 329,
00369 uid_1_2_840_10008_5_1_4_1_1_78_2 = 330,
00370 uid_1_2_840_10008_5_1_4_1_1_78_3 = 331,
00371 uid_1_2_840_10008_5_1_4_1_1_78_4 = 332,
00372 uid_1_2_840_10008_5_1_4_1_1_78_5 = 333,
00373 uid_1_2_840_10008_5_1_4_1_1_78_6 = 334,
00374 uid_1_2_840_10008_5_1_4_1_1_78_7 = 335,
00375 uid_1_2_840_10008_5_1_4_1_1_78_8 = 336,
00376 uid_1_2_840_10008_5_1_4_1_1_79_1 = 337,
00377 uid_1_2_840_10008_5_1_4_1_1_80_1 = 338,
00378 uid_1_2_840_10008_5_1_4_1_1_81_1 = 339,
00379 uid_1_2_840_10008_5_1_4_1_1_82_1 = 340,
00380 uid_1_2_840_10008_5_1_4_1_1_88_34 = 341,
00381 uid_1_2_840_10008_5_1_4_1_1_88_35 = 342,
00382 uid_1_2_840_10008_5_1_4_1_1_88_68 = 343,

```

```

00383 uid_1_2_840_10008_5_1_4_1_1_88_69 = 344,
00384 uid_1_2_840_10008_5_1_4_1_1_88_70 = 345,
00385 uid_1_2_840_10008_5_1_4_1_1_88_71 = 346,
00386 uid_1_2_840_10008_5_1_4_1_1_88_72 = 347,
00387 uid_1_2_840_10008_5_1_4_1_1_88_73 = 348,
00388 uid_1_2_840_10008_5_1_4_1_1_88_74 = 349,
00389 uid_1_2_840_10008_5_1_4_1_1_88_75 = 350,
00390 uid_1_2_840_10008_5_1_4_1_1_90_1 = 351,
00391 uid_1_2_840_10008_5_1_4_1_1_104_3 = 352,
00392 uid_1_2_840_10008_5_1_4_1_1_130 = 353,
00393 uid_1_2_840_10008_5_1_4_1_1_131 = 354,
00394 uid_1_2_840_10008_5_1_4_1_1_200_1 = 355,
00395 uid_1_2_840_10008_5_1_4_1_1_200_2 = 356,
00396 uid_1_2_840_10008_5_1_4_1_1_200_3 = 357,
00397 uid_1_2_840_10008_5_1_4_1_1_200_4 = 358,
00398 uid_1_2_840_10008_5_1_4_1_1_200_5 = 359,
00399 uid_1_2_840_10008_5_1_4_1_1_200_6 = 360,
00400 uid_1_2_840_10008_5_1_4_1_1_481_10 = 361,
00401 uid_1_2_840_10008_5_1_4_1_1_481_11 = 362,
00402 uid_1_2_840_10008_5_1_4_1_1_501_1 = 363,
00403 uid_1_2_840_10008_5_1_4_1_1_501_2_1 = 364,
00404 uid_1_2_840_10008_5_1_4_1_1_501_2_2 = 365,
00405 uid_1_2_840_10008_5_1_4_1_1_501_3 = 366,
00406 uid_1_2_840_10008_5_1_4_1_1_501_4 = 367,
00407 uid_1_2_840_10008_5_1_4_1_1_501_5 = 368,
00408 uid_1_2_840_10008_5_1_4_1_1_501_6 = 369,
00409 uid_1_2_840_10008_5_1_4_1_1_601_1 = 370,
00410 uid_1_2_840_10008_5_1_4_1_1_601_2 = 371,
00411 uid_1_2_840_10008_5_1_4_1_2_4_2 = 372,
00412 uid_1_2_840_10008_5_1_4_1_2_4_3 = 373,
00413 uid_1_2_840_10008_5_1_4_1_2_5_3 = 374,
00414 uid_1_2_840_10008_5_1_4_20_1 = 375,
00415 uid_1_2_840_10008_5_1_4_20_2 = 376,
00416 uid_1_2_840_10008_5_1_4_20_3 = 377,
00417 uid_1_2_840_10008_5_1_4_34_5_1 = 378,
00418 uid_1_2_840_10008_5_1_4_34_6 = 379,
00419 uid_1_2_840_10008_5_1_4_34_6_1 = 380,
00420 uid_1_2_840_10008_5_1_4_34_6_2 = 381,
00421 uid_1_2_840_10008_5_1_4_34_6_3 = 382,
00422 uid_1_2_840_10008_5_1_4_34_6_4 = 383,
00423 uid_1_2_840_10008_5_1_4_34_7 = 384,
00424 uid_1_2_840_10008_5_1_4_34_8 = 385,
00425 uid_1_2_840_10008_5_1_4_34_9 = 386,
00426 uid_1_2_840_10008_5_1_4_34_10 = 387,
00427 uid_1_2_840_10008_5_1_4_38_4 = 388,
00428 uid_1_2_840_10008_5_1_4_39_1 = 389,
00429 uid_1_2_840_10008_5_1_4_39_2 = 390,
00430 uid_1_2_840_10008_5_1_4_39_3 = 391,
00431 uid_1_2_840_10008_5_1_4_39_4 = 392,
00432 uid_1_2_840_10008_5_1_4_43_1 = 393,
00433 uid_1_2_840_10008_5_1_4_43_2 = 394,
00434 uid_1_2_840_10008_5_1_4_43_3 = 395,
00435 uid_1_2_840_10008_5_1_4_43_4 = 396,
00436 uid_1_2_840_10008_5_1_4_44_1 = 397,
00437 uid_1_2_840_10008_5_1_4_44_2 = 398,
00438 uid_1_2_840_10008_5_1_4_44_3 = 399,
00439 uid_1_2_840_10008_5_1_4_44_4 = 400,
00440 uid_1_2_840_10008_5_1_4_45_1 = 401,
00441 uid_1_2_840_10008_5_1_4_45_2 = 402,
00442 uid_1_2_840_10008_5_1_4_45_3 = 403,
00443 uid_1_2_840_10008_5_1_4_45_4 = 404,
00444 uid_1_2_840_10008_7_1_1 = 405,
00445 uid_1_2_840_10008_7_1_2 = 406,
00446 uid_1_2_840_10008_8_1_1 = 407,
00447 uid_1_2_840_10008_5_1_4_1_1_4_3 = 408,
00448 uid_1_2_840_10008_15_1_1 = 409
00449 //
00450 //
00452
00454 //
00455 // Optionally private UIDs
00456 //
00457 #if 0
00458 uid_1_2_840_113619_4_2,
00459 uid_1_2_840_113619_4_3,
00460 uid_1_3_12_2_1107_5_9_1,
00461 uid_1_2_840_113619_4_26,
00462 uid_1_2_840_113619_4_30,
00463 uid_2_16_840_1_113709_1_5_1,
00464 uid_2_16_840_1_113709_1_2_2,
00465 uid_1_2_840_113543_6_6_1_3_10002,

```



```

00466 uid_1_2_392_200036_9116_7_8_1_1_1,
00467 uid_1_2_392_200036_9125_1_1_2,
00468 uid_1_2_840_113619_4_27,
00469 uid_1_3_46_670589_11_0_0_12_1,
00470 uid_1_3_46_670589_11_0_0_12_2,
00471 uid_1_3_46_670589_11_0_0_12_4,
00472 uid_1_3_46_670589_2_3_1_1,
00473 uid_1_3_46_670589_2_4_1_1,
00474 uid_1_3_46_670589_2_5_1_1,
00475 uid_1_3_46_670589_5_0_1,
00476 uid_1_3_46_670589_5_0_1_1,
00477 uid_1_3_46_670589_5_0_10,
00478 uid_1_3_46_670589_5_0_11,
00479 uid_1_3_46_670589_5_0_11_1,
00480 uid_1_3_46_670589_5_0_12,
00481 uid_1_3_46_670589_5_0_13,
00482 uid_1_3_46_670589_5_0_14,
00483 uid_1_3_46_670589_5_0_2,
00484 uid_1_3_46_670589_5_0_2_1,
00485 uid_1_3_46_670589_5_0_3,
00486 uid_1_3_46_670589_5_0_3_1,
00487 uid_1_3_46_670589_5_0_4,
00488 uid_1_3_46_670589_5_0_7,
00489 uid_1_3_46_670589_5_0_8,
00490 uid_1_3_46_670589_5_0_9,
00491 uid_1_2_752_24_3_7_6,
00492 uid_1_2_752_24_3_7_7,
00493 uid_1_2_840_113619_5_2,
00494 uid_1_3_46_670589_33_1_4_1
00495 #endif
00496 //
00497 //
00499
00500 } TSType;
00501 typedef enum {
00502 VerificationSOPClass = 1, // Verification SOP Class
00503 ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2, // Implicit VR Little Endian: Default Transfer Syntax for
DICOM
00504 ExplicitVRLittleEndian = 3, // Explicit VR Little Endian
00505 DeflatedExplicitVRLittleEndian = 4, // Deflated Explicit VR Little Endian
00506 ExplicitVRBigEndian = 5, // Explicit VR Big Endian
00507 JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6, // JPEG Baseline (Process 1): Default
Transfer Syntax for Lossy JPEG 8 Bit Image Compression
00508 JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7, // JPEG Extended
(Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)
00509 JPEGExtendedProcess35Retired = 8, // JPEG Extended (Process 3 & 5)
00510 JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
00511 JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
00512 JPEGFullProgressionNonHierarchicalProcess1012Retired = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
00513 JPEGFullProgressionNonHierarchicalProcess1113Retired = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
00514 JPEGLosslessNonHierarchicalProcess14 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
00515 JPEGLosslessNonHierarchicalProcess15Retired = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
00516 JPEGExtendedHierarchicalProcess1618Retired = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
00517 JPEGExtendedHierarchicalProcess1719Retired = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
00518 JPEGSpectralSelectionHierarchicalProcess2022Retired = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
00519 JPEGSpectralSelectionHierarchicalProcess2123Retired = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
00520 JPEGFullProgressionHierarchicalProcess2426Retired = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
00521 JPEGFullProgressionHierarchicalProcess2527Retired = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
00522 JPEGLosslessHierarchicalProcess28Retired = 21, // JPEG Lossless, Hierarchical (Process 28)
00523 JPEGLosslessHierarchicalProcess29Retired = 22, // JPEG Lossless, Hierarchical (Process 29)
00524
    JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression
    = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]): Default Transfer Syntax for
    Lossless JPEG Image Compression
00525 JPEGLSLosslessImageCompression = 24, // JPEG-LS Lossless Image Compression
00526 JPEGLSLossyNearLosslessImageCompression = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
00527 JPEG2000ImageCompressionLosslessOnly = 26, // JPEG 2000 Image Compression (Lossless Only)
00528 JPEG2000ImageCompression = 27, // JPEG 2000 Image Compression
00529 JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28, // JPEG 2000 Part 2 Multi-component Image
    Compression (Lossless Only)
00530 JPEG2000Part2MulticomponentImageCompression = 29, // JPEG 2000 Part 2 Multi-component Image Compression
00531 JPIPReferenced = 30, // JPIP Referenced
00532 JPIPReferencedDeflate = 31, // JPIP Referenced Deflate
00533 MPEG2MainProfileMainLevel = 32, // MPEG2 Main Profile @ Main Level
00534 RLELossless = 33, // RLE Lossless
00535 RFC2557MIMEencapsulation = 34, // RFC 2557 MIME encapsulation
00536 XMLEncoding = 35, // XML Encoding
00537 MediaStorageDirectoryStorage = 36, // Media Storage Directory Storage
00538 TalairachBrainAtlasFrameofReference = 37, // Talairach Brain Atlas Frame of Reference
00539 SPM2T1FrameofReference = 38, // SPM2 T1 Frame of Reference
00540 SPM2T2FrameofReference = 39, // SPM2 T2 Frame of Reference

```

```
00541 SPM2PDFFrameofReference = 40, // SPM2 PD Frame of Reference
00542 SPM2EPIFrameofReference = 41, // SPM2 EPI Frame of Reference
00543 SPM2FILTT1FrameofReference = 42, // SPM2 FIL T1 Frame of Reference
00544 SPM2PETFrameofReference = 43, // SPM2 PET Frame of Reference
00545 SPM2TRANSMFrameofReference = 44, // SPM2 TRANSM Frame of Reference
00546 SPM2SPECTFrameofReference = 45, // SPM2 SPECT Frame of Reference
00547 SPM2GRAYFrameofReference = 46, // SPM2 GRAY Frame of Reference
00548 SPM2WHITEFrameofReference = 47, // SPM2 WHITE Frame of Reference
00549 SPM2CSFFrameofReference = 48, // SPM2 CSF Frame of Reference
00550 SPM2BRAINMASKFrameofReference = 49, // SPM2 BRAINMASK Frame of Reference
00551 SPM2AVG305T1FrameofReference = 50, // SPM2 AVG305T1 Frame of Reference
00552 SPM2AVG152T1FrameofReference = 51, // SPM2 AVG152T1 Frame of Reference
00553 SPM2AVG152T2FrameofReference = 52, // SPM2 AVG152T2 Frame of Reference
00554 SPM2AVG152PDFFrameofReference = 53, // SPM2 AVG152PD Frame of Reference
00555 SPM2SINGLESUBJT1FrameofReference = 54, // SPM2 SINGLESUBJT1 Frame of Reference
00556 ICBM452T1FrameofReference = 55, // ICBM 452 T1 Frame of Reference
00557 ICBMSingleSubjectMRIFrameofReference = 56, // ICBM Single Subject MRI Frame of Reference
00558 BasicStudyContentNotificationSOPClassRetired = 57, // Basic Study Content Notification SOP Class
00559 StorageCommitmentPushModelSOPClass = 58, // Storage Commitment Push Model SOP Class
00560 StorageCommitmentPushModelSOPInstance = 59, // Storage Commitment Push Model SOP Instance
00561 StorageCommitmentPullModelSOPClassRetired = 60, // Storage Commitment Pull Model SOP Class
00562 StorageCommitmentPullModelSOPInstanceRetired = 61, // Storage Commitment Pull Model SOP Instance
00563 ProceduralEventLoggingSOPClass = 62, // Procedural Event Logging SOP Class
00564 ProceduralEventLoggingSOPInstance = 63, // Procedural Event Logging SOP Instance
00565 SubstanceAdministrationLoggingSOPClass = 64, // Substance Administration Logging SOP Class
00566 SubstanceAdministrationLoggingSOPInstance = 65, // Substance Administration Logging SOP Instance
00567 DICOMUIDRegistry = 66, // DICOM UID Registry
00568 DICOMControlledTerminology = 67, // DICOM Controlled Terminology
00569 DICOMApplicationContextName = 68, // DICOM Application Context Name
00570 DetachedPatientManagementSOPClassRetired = 69, // Detached Patient Management SOP Class
00571 DetachedPatientManagementMetaSOPClassRetired = 70, // Detached Patient Management Meta SOP Class
00572 DetachedVisitManagementSOPClassRetired = 71, // Detached Visit Management SOP Class
00573 DetachedStudyManagementSOPClassRetired = 72, // Detached Study Management SOP Class
00574 StudyComponentManagementSOPClassRetired = 73, // Study Component Management SOP Class
00575 ModalityPerformedProcedureStepSOPClass = 74, // Modality Performed Procedure Step SOP Class
00576 ModalityPerformedProcedureStepRetrieveSOPClass = 75, // Modality Performed Procedure Step Retrieve SOP Class
00577 ModalityPerformedProcedureStepNotificationSOPClass = 76, // Modality Performed Procedure Step Notification SOP Class
00578 DetachedResultsManagementSOPClassRetired = 77, // Detached Results Management SOP Class
00579 DetachedResultsManagementMetaSOPClassRetired = 78, // Detached Results Management Meta SOP Class
00580 DetachedStudyManagementMetaSOPClassRetired = 79, // Detached Study Management Meta SOP Class
00581 DetachedInterpretationManagementSOPClassRetired = 80, // Detached Interpretation Management SOP Class
00582 StorageServiceClass = 81, // Storage Service Class
00583 BasicFilmSessionSOPClass = 82, // Basic Film Session SOP Class
00584 BasicFilmBoxSOPClass = 83, // Basic Film Box SOP Class
00585 BasicGrayscaleImageBoxSOPClass = 84, // Basic Grayscale Image Box SOP Class
00586 BasicColorImageBoxSOPClass = 85, // Basic Color Image Box SOP Class
00587 ReferencedImageBoxSOPClassRetired = 86, // Referenced Image Box SOP Class
00588 BasicGrayscalePrintManagementMetaSOPClass = 87, // Basic Grayscale Print Management Meta SOP Class
00589 ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88, // Referenced Grayscale Print Management Meta SOP Class
00590 PrintJobSOPClass = 89, // Print Job SOP Class
00591 BasicAnnotationBoxSOPClass = 90, // Basic Annotation Box SOP Class
00592 PrinterSOPClass = 91, // Printer SOP Class
00593 PrinterConfigurationRetrievalSOPClass = 92, // Printer Configuration Retrieval SOP Class
00594 PrinterSOPInstance = 93, // Printer SOP Instance
00595 PrinterConfigurationRetrievalSOPInstance = 94, // Printer Configuration Retrieval SOP Instance
00596 BasicColorPrintManagementMetaSOPClass = 95, // Basic Color Print Management Meta SOP Class
00597 ReferencedColorPrintManagementMetaSOPClassRetired = 96, // Referenced Color Print Management Meta SOP Class
00598 VOILUTBoxSOPClass = 97, // VOI LUT Box SOP Class
00599 PresentationLUTSOPClass = 98, // Presentation LUT SOP Class
00600 ImageOverlayBoxSOPClassRetired = 99, // Image Overlay Box SOP Class
00601 BasicPrintImageOverlayBoxSOPClassRetired = 100, // Basic Print Image Overlay Box SOP Class
00602 PrintQueueSOPInstanceRetired = 101, // Print Queue SOP Instance
00603 PrintQueueManagementSOPClassRetired = 102, // Print Queue Management SOP Class
00604 StoredPrintStorageSOPClassRetired = 103, // Stored Print Storage SOP Class
00605 HardcopyGrayscaleImageStorageSOPClassRetired = 104, // Hardcopy Grayscale Image Storage SOP Class
00606 HardcopyColorImageStorageSOPClassRetired = 105, // Hardcopy Color Image Storage SOP Class
00607 PullPrintRequestSOPClassRetired = 106, // Pull Print Request SOP Class
00608 PullStoredPrintManagementMetaSOPClassRetired = 107, // Pull Stored Print Management Meta SOP Class
00609 MediaCreationManagementSOPClassUID = 108, // Media Creation Management SOP Class UID
00610 ComputedRadiographyImageStorage = 109, // Computed Radiography Image Storage
00611 DigitalXRayImageStorageForPresentation = 110, // Digital X-Ray Image Storage - For Presentation
00612 DigitalXRayImageStorageForProcessing = 111, // Digital X-Ray Image Storage - For Processing
00613 DigitalMammographyXRayImageStorageForPresentation = 112, // Digital Mammography X-Ray Image Storage - For
Presentation
00614 DigitalMammographyXRayImageStorageForProcessing = 113, // Digital Mammography X-Ray Image Storage - For Processing
00615 DigitalIntraoralXRayImageStorageForPresentation = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
00616 DigitalIntraoralXRayImageStorageForProcessing = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
00617 CTImageStorage = 116, // CT Image Storage
00618 EnhancedCTImageStorage = 117, // Enhanced CT Image Storage
00619 UltrasoundMultiframeImageStorageRetired = 118, // Ultrasound Multi-frame Image Storage
00620 UltrasoundMultiframeImageStorage = 119, // Ultrasound Multi-frame Image Storage
```

```
00621 MRImageStorage = 120, // MR Image Storage
00622 EnhancedMRImageStorage = 121, // Enhanced MR Image Storage
00623 MRSpectroscopyStorage = 122, // MR Spectroscopy Storage
00624 NuclearMedicineImageStorageRetired = 123, // Nuclear Medicine Image Storage
00625 UltrasoundImageStorageRetired = 124, // Ultrasound Image Storage
00626 UltrasoundImageStorage = 125, // Ultrasound Image Storage
00627 SecondaryCaptureImageStorage = 126, // Secondary Capture Image Storage
00628 MultiframeSingleBitSecondaryCaptureImageStorage = 127, // Multi-frame Single Bit Secondary Capture Image Storage
00629 MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128, // Multi-frame Grayscale Byte Secondary Capture Image
Storage
00630 MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129, // Multi-frame Grayscale Word Secondary Capture Image
Storage
00631 MultiframeTrueColorSecondaryCaptureImageStorage = 130, // Multi-frame True Color Secondary Capture Image Storage
00632 StandaloneOverlayStorageRetired = 131, // Standalone Overlay Storage
00633 StandaloneCurveStorageRetired = 132, // Standalone Curve Storage
00634 WaveformStorageTrialRetired = 133, // Waveform Storage - Trial
00635 ECG12leadWaveformStorage = 134, // 12-lead ECG Waveform Storage
00636 GeneralECGWaveformStorage = 135, // General ECG Waveform Storage
00637 AmbulatoryECGWaveformStorage = 136, // Ambulatory ECG Waveform Storage
00638 HemodynamicWaveformStorage = 137, // Hemodynamic Waveform Storage
00639 CardiacElectrophysiologyWaveformStorage = 138, // Cardiac Electrophysiology Waveform Storage
00640 BasicVoiceAudioWaveformStorage = 139, // Basic Voice Audio Waveform Storage
00641 StandaloneModalityLUTStorageRetired = 140, // Standalone Modality LUT Storage
00642 StandaloneVOILUTStorageRetired = 141, // Standalone VOI LUT Storage
00643 GrayscaleSoftcopyPresentationStateStorageSOPClass = 142, // Grayscale Softcopy Presentation State Storage SOP Class
00644 ColorSoftcopyPresentationStateStorageSOPClass = 143, // Color Softcopy Presentation State Storage SOP Class
00645 PseudoColorSoftcopyPresentationStateStorageSOPClass = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
00646 BlendingSoftcopyPresentationStateStorageSOPClass = 145, // Blending Softcopy Presentation State Storage SOP Class
00647 XRayAngiographicImageStorage = 146, // X-Ray Angiographic Image Storage
00648 EnhancedXAImageStorage = 147, // Enhanced XA Image Storage
00649 XRayRadiofluoroscopicImageStorage = 148, // X-Ray Radiofluoroscopic Image Storage
00650 EnhancedXRFImageStorage = 149, // Enhanced XRF Image Storage
00651 XRay3DAngiographicImageStorage = 150, // X-Ray 3D Angiographic Image Storage
00652 XRay3DCraniofacialImageStorage = 151, // X-Ray 3D Craniofacial Image Storage
00653 XRayAngiographicBiPlaneImageStorageRetired = 152, // X-Ray Angiographic Bi-Plane Image Storage
00654 NuclearMedicineImageStorage = 153, // Nuclear Medicine Image Storage
00655 RawDataStorage = 154, // Raw Data Storage
00656 SpatialRegistrationStorage = 155, // Spatial Registration Storage
00657 SpatialFiducialsStorage = 156, // Spatial Fiducials Storage
00658 DeformableSpatialRegistrationStorage = 157, // Deformable Spatial Registration Storage
00659 SegmentationStorage = 158, // Segmentation Storage
00660 RealWorldValueMappingStorage = 159, // Real World Value Mapping Storage
00661 VLImageStorageTrialRetired = 160, // VL Image Storage - Trial
00662 VLMultiframeImageStorageTrialRetired = 161, // VL Multi-frame Image Storage - Trial
00663 VLEndoscopicImageStorage = 162, // VL Endoscopic Image Storage
00664 VideoEndoscopicImageStorage = 163, // Video Endoscopic Image Storage
00665 VLMicroscopicImageStorage = 164, // VL Microscopic Image Storage
00666 VideoMicroscopicImageStorage = 165, // Video Microscopic Image Storage
00667 VLSlideCoordinatesMicroscopicImageStorage = 166, // VL Slide-Coordinates Microscopic Image Storage
00668 VLPhotographicImageStorage = 167, // VL Photographic Image Storage
00669 VideoPhotographicImageStorage = 168, // Video Photographic Image Storage
00670 OphthalmicPhotography8BitImageStorage = 169, // Ophthalmic Photography 8 Bit Image Storage
00671 OphthalmicPhotography16BitImageStorage = 170, // Ophthalmic Photography 16 Bit Image Storage
00672 StereometricRelationshipStorage = 171, // Stereometric Relationship Storage
00673 OphthalmicTomographyImageStorage = 172, // Ophthalmic Tomography Image Storage
00674 TextSRStorageTrialRetired = 173, // Text SR Storage - Trial
00675 AudioSRStorageTrialRetired = 174, // Audio SR Storage - Trial
00676 DetailSRStorageTrialRetired = 175, // Detail SR Storage - Trial
00677 ComprehensiveSRStorageTrialRetired = 176, // Comprehensive SR Storage - Trial
00678 BasicTextSRStorage = 177, // Basic Text SR Storage
00679 EnhancedSRStorage = 178, // Enhanced SR Storage
00680 ComprehensiveSRStorage = 179, // Comprehensive SR Storage
00681 ProcedureLogStorage = 180, // Procedure Log Storage
00682 MammographyCADSRStorage = 181, // Mammography CAD SR Storage
00683 KeyObjectSelectionDocumentStorage = 182, // Key Object Selection Document Storage
00684 ChestCADSRStorage = 183, // Chest CAD SR Storage
00685 XRayRadiationDoseSRStorage = 184, // X-Ray Radiation Dose SR Storage
00686 EncapsulatedPDFStorage = 185, // Encapsulated PDF Storage
00687 EncapsulatedCDAStorage = 186, // Encapsulated CDA Storage
00688 PositronEmissionTomographyImageStorage = 187, // Positron Emission Tomography Image Storage
00689 StandalonePETCurveStorageRetired = 188, // Standalone PET Curve Storage
00690 RTImageStorage = 189, // RT Image Storage
00691 RTDoseStorage = 190, // RT Dose Storage
00692 RTStructureSetStorage = 191, // RT Structure Set Storage
00693 RTBeamsTreatmentRecordStorage = 192, // RT Beams Treatment Record Storage
00694 RTPlanStorage = 193, // RT Plan Storage
00695 RTBrachyTreatmentRecordStorage = 194, // RT Brachy Treatment Record Storage
00696 RTTreatmentSummaryRecordStorage = 195, // RT Treatment Summary Record Storage
00697 RTIonPlanStorage = 196, // RT Ion Plan Storage
00698 RTIonBeamsTreatmentRecordStorage = 197, // RT Ion Beams Treatment Record Storage
00699 PatientRootQueryRetrieveInformationModelFIND = 198, // Patient Root Query/Retrieve Information Model - FIND
```

```

00700 PatientRootQueryRetrieveInformationModelMOVE = 199, // Patient Root Query/Retrieve Information Model - MOVE
00701 PatientRootQueryRetrieveInformationModelGET = 200, // Patient Root Query/Retrieve Information Model - GET
00702 StudyRootQueryRetrieveInformationModelFIND = 201, // Study Root Query/Retrieve Information Model - FIND
00703 StudyRootQueryRetrieveInformationModelMOVE = 202, // Study Root Query/Retrieve Information Model - MOVE
00704 StudyRootQueryRetrieveInformationModelGET = 203, // Study Root Query/Retrieve Information Model - GET
00705 PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204, // Patient/Study Only Query/Retrieve Information
      Model - FIND
00706 PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205, // Patient/Study Only Query/Retrieve Information
      Model - MOVE
00707 PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206, // Patient/Study Only Query/Retrieve Information
      Model - GET
00708 ModalityWorklistInformationModelFIND = 207, // Modality Worklist Information Model - FIND
00709 GeneralPurposeWorklistInformationModelFIND = 208, // General Purpose Worklist Information Model - FIND
00710 GeneralPurposeScheduledProcedureStepSOPClass = 209, // General Purpose Scheduled Procedure Step SOP Class
00711 GeneralPurposePerformedProcedureStepSOPClass = 210, // General Purpose Performed Procedure Step SOP Class
00712 GeneralPurposeWorklistManagementMetaSOPClass = 211, // General Purpose Worklist Management Meta SOP Class
00713 InstanceAvailabilityNotificationSOPClass = 212, // Instance Availability Notification SOP Class
00714 RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213, // RT Beams Delivery Instruction Storage (Supplement
      74 Frozen Draft)
00715 RTConventionalMachineVerificationSupplement74FrozenDraft = 214, // RT Conventional Machine Verification (Supplement 74
      Frozen Draft)
00716 RTIonMachineVerificationSupplement74FrozenDraft = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
00717 UnifiedWorklistandProcedureStepServiceClass = 216, // Unified Worklist and Procedure Step Service Class
00718 UnifiedProcedureStepPushSOPClass = 217, // Unified Procedure Step - Push SOP Class
00719 UnifiedProcedureStepWatchSOPClass = 218, // Unified Procedure Step - Watch SOP Class
00720 UnifiedProcedureStepPullSOPClass = 219, // Unified Procedure Step - Pull SOP Class
00721 UnifiedProcedureStepEventSOPClass = 220, // Unified Procedure Step - Event SOP Class
00722 UnifiedWorklistandProcedureStepSOPInstance = 221, // Unified Worklist and Procedure Step SOP Instance
00723 GeneralRelevantPatientInformationQuery = 222, // General Relevant Patient Information Query
00724 BreastImagingRelevantPatientInformationQuery = 223, // Breast Imaging Relevant Patient Information Query
00725 CardiacRelevantPatientInformationQuery = 224, // Cardiac Relevant Patient Information Query
00726 HangingProtocolStorage = 225, // Hanging Protocol Storage
00727 HangingProtocolInformationModelFIND = 226, // Hanging Protocol Information Model - FIND
00728 HangingProtocolInformationModelMOVE = 227, // Hanging Protocol Information Model - MOVE
00729 ProductCharacteristicsQuerySOPClass = 228, // Product Characteristics Query SOP Class
00730 SubstanceApprovalQuerySOPClass = 229, // Substance Approval Query SOP Class
00731 dicomDeviceName = 230, // dicomDeviceName
00732 dicomDescription = 231, // dicomDescription
00733 dicomManufacturer = 232, // dicomManufacturer
00734 dicomManufacturerModelName = 233, // dicomManufacturerModelName
00735 dicomSoftwareVersion = 234, // dicomSoftwareVersion
00736 dicomVendorData = 235, // dicomVendorData
00737 dicomAETitle = 236, // dicomAETitle
00738 dicomNetworkConnectionReference = 237, // dicomNetworkConnectionReference
00739 dicomApplicationCluster = 238, // dicomApplicationCluster
00740 dicomAssociationInitiator = 239, // dicomAssociationInitiator
00741 dicomAssociationAcceptor = 240, // dicomAssociationAcceptor
00742 dicomHostname = 241, // dicomHostname
00743 dicomPort = 242, // dicomPort
00744 dicomSOPClass = 243, // dicomSOPClass
00745 dicomTransferRole = 244, // dicomTransferRole
00746 dicomTransferSyntax = 245, // dicomTransferSyntax
00747 dicomPrimaryDeviceType = 246, // dicomPrimaryDeviceType
00748 dicomRelatedDeviceReference = 247, // dicomRelatedDeviceReference
00749 dicomPreferredCalledAETitle = 248, // dicomPreferredCalledAETitle
00750 dicomTLSCyphersuite = 249, // dicomTLSCyphersuite
00751 dicomAuthorizedNodeCertificateReference = 250, // dicomAuthorizedNodeCertificateReference
00752 dicomThisNodeCertificateReference = 251, // dicomThisNodeCertificateReference
00753 dicomInstalled = 252, // dicomInstalled
00754 dicomStationName = 253, // dicomStationName
00755 dicomDeviceSerialNumber = 254, // dicomDeviceSerialNumber
00756 dicomInstitutionName = 255, // dicomInstitutionName
00757 dicomInstitutionAddress = 256, // dicomInstitutionAddress
00758 dicomInstitutionDepartmentName = 257, // dicomInstitutionDepartmentName
00759 dicomIssuerOfPatientID = 258, // dicomIssuerOfPatientID
00760 dicomPreferredCallingAETitle = 259, // dicomPreferredCallingAETitle
00761 dicomSupportedCharacterSet = 260, // dicomSupportedCharacterSet
00762 dicomConfigurationRoot = 261, // dicomConfigurationRoot
00763 dicomDevicesRoot = 262, // dicomDevicesRoot
00764 dicomUniqueAETitlesRegistryRoot = 263, // dicomUniqueAETitlesRegistryRoot
00765 dicomDevice = 264, // dicomDevice
00766 dicomNetworkAE = 265, // dicomNetworkAE
00767 dicomNetworkConnection = 266, // dicomNetworkConnection
00768 dicomUniqueAETitle = 267, // dicomUniqueAETitle
00769 dicomTransferCapability = 268, // dicomTransferCapability
00770 //
00771 VLWholeSlideMicroscopyImageStorage = 269,
00772 EnhancedUSVolumeStorage = 270,
00773 SurfaceSegmentationStorage = 271,
00774 BreastTomosynthesisImageStorage = 272,
00775 LegacyConvertedEnhancedCTImageStorage = 273,

```



```

00776 LegacyConvertedEnhancedMRIImageStorage      = 274,
00777 LegacyConvertedEnhancedPETImageStorage        = 275,
00778 MPEG2MainProfileHighLevel                     = 276,
00779 MPEG4AVCH_264HighProfileLevel4_1              = 277,
00780 MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278,
00781
00783 //
00784 // 2019b
00785 //
00786 PETColorPaletteSOPInstance                     = 279,
00787 HotMetalBlueColorPaletteSOPInstance            = 280,
00788 PET20StepColorPaletteSOPInstance              = 281,
00789 SpringColorPaletteSOPInstance                 = 282,
00790 SummerColorPaletteSOPInstance                 = 283,
00791 FallColorPaletteSOPInstance                   = 284,
00792 WinterColorPaletteSOPInstance                 = 285,
00793 Papyrus3ImplicitVRLittleEndian                = 286,
00794 AdultMouseAnatomyOntology                     = 287,
00795 UberonOntology                               = 288,
00796 IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN = 289,
00797 MouseGenomeInitiativeMGI                     = 290,
00798 PubChemCompoundCID                           = 291,
00799 ICD11                                         = 292,
00800 NewYorkUniversityMelanomaClinicalCooperativeGroup = 293,
00801 MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide = 294,
00802 ImageBiomarkerStandardisationInitiative      = 295,
00803 RadiomicsOntology                           = 296,
00804 DisplaySystemSOPClass                       = 297,
00805 DisplaySystemSOPInstance                     = 298,
00806 GeneralAudioWaveformStorage                 = 299,
00807 ArterialPulseWaveformStorage                 = 300,
00808 RespiratoryWaveformStorage                   = 301,
00809 XAXRFGrayscaleSoftcopyPresentationStateStorage = 302,
00810 GrayscalePlanarMPRVolumetricPresentationStateStorage = 303,
00811 MPEG4AVCH_264HighProfileLevel4_2For2DVideo = 304,
00812 MPEG4AVCH_264HighProfileLevel4_2For3DVideo = 305,
00813 MPEG4AVCH_264StereoHighProfileLevel4_2      = 306,
00814 HEVCH_265MainProfileLevel5_1                = 307,
00815 HEVCH_265Main10ProfileLevel5_1             = 308,
00816 HotIronColorPaletteSOPInstance              = 309,
00817 CompositingPlanarMPRVolumetricPresentationStateStorage = 310,
00818 AdvancedBlendingPresentationStateStorage     = 311,
00819 VolumeRenderingVolumetricPresentationStateStorage = 312,
00820 SegmentedVolumeRenderingVolumetricPresentationStateStorage = 313,
00821 MultipleVolumeRenderingVolumetricPresentationStateStorage = 314,
00822 Null0                                         = 315,
00823 BreastProjectionXRayImageStorageForPresentation = 316,
00824 BreastProjectionXRayImageStorageForProcessing = 317,
00825 IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318,
00826 IntravascularOpticalCoherenceTomographyImageStorageForProcessing = 319,
00827 ParametricMapStorage                        = 320,
00828 Null1                                         = 321,
00829 TractographyResultsStorage                   = 322,
00830 SurfaceScanMeshStorage                       = 323,
00831 SurfaceScanPointCloudStorage                 = 324,
00832 WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325,
00833 WideFieldOphthalmicPhotography3DCoordinatesImageStorage = 326,
00834 OphthalmicOpticalCoherenceTomographyEnFaceImageStorage = 327,
00835 OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage = 328,
00836 LensometryMeasurementsStorage               = 329,
00837 AutorefractionMeasurementsStorage           = 330,
00838 KeratometryMeasurementsStorage              = 331,
00839 SubjectiveRefractionMeasurementsStorage     = 332,
00840 VisualAcuityMeasurementsStorage             = 333,
00841 SpectaclePrescriptionReportStorage           = 334,
00842 OphthalmicAxialMeasurementsStorage          = 335,
00843 IntraocularLensCalculationsStorage           = 336,
00844 MacularGridThicknessandVolumeReportStorage = 337,
00845 OphthalmicVisualFieldStaticPerimetryMeasurementsStorage = 338,
00846 OphthalmicThicknessMapStorage               = 339,
00847 CornealTopographyMapStorage                 = 340,
00848 Comprehensive3DSRStorage                    = 341,
00849 ExtensibleSRStorage                         = 342,
00850 RadiopharmaceuticalRadiationDoseSRStorage   = 343,
00851 ColonCADSRStorage                           = 344,
00852 ImplantationPlanSRStorage                   = 345,
00853 AcquisitionContextSRStorage                 = 346,
00854 SimplifiedAdultEchoSRStorage                = 347,
00855 PatientRadiationDoseSRStorage               = 348,
00856 PlannedImagingAgentAdministrationSRStorage = 349,
00857 PerformedImagingAgentAdministrationSRStorage = 350,

```

```

00858 ContentAssessmentResultsStorage           = 351,
00859 EncapsulatedSTLStorage                     = 352,
00860 EnhancedPETImageStorage                     = 353,
00861 BasicStructuredDisplayStorage               = 354,
00862 CTDefinedProcedureProtocolStorage           = 355,
00863 CTPerformedProcedureProtocolStorage         = 356,
00864 ProtocolApprovalStorage                     = 357,
00865 ProtocolApprovalInformationModelFIND        = 358,
00866 ProtocolApprovalInformationModelMOVE        = 359,
00867 ProtocolApprovalInformationModelGET         = 360,
00868 RTPhysicianIntentStorage                    = 361,
00869 RTSegmentAnnotationStorage                 = 362,
00870 DICOSCTImageStorage                         = 363,
00871 DICOSDigitalXRayImageStorageForPresentation = 364,
00872 DICOSDigitalXRayImageStorageForProcessing  = 365,
00873 DICOSThreatDetectionReportStorage          = 366,
00874 DICOS2DAITStorage                          = 367,
00875 DICOS3DAITStorage                          = 368,
00876 DICOSQuadrupoleResonanceQRStorage         = 369,
00877 EddyCurrentImageStorage                    = 370,
00878 EddyCurrentMultiframeImageStorage          = 371,
00879 CompositeInstanceRootRetrieveMOVE           = 372,
00880 CompositeInstanceRootRetrieveGET            = 373,
00881 CompositeInstanceRetrieveWithoutBulkDataGET = 374,
00882 DefinedProcedureProtocolInformationModelFIND = 375,
00883 DefinedProcedureProtocolInformationModelMOVE = 376,
00884 DefinedProcedureProtocolInformationModelGET  = 377,
00885 UPSFilteredGlobalSubscriptionSOPInstance    = 378,
00886 UnifiedWorklistandProcedureStepServiceClass1 = 379,
00887 UnifiedProcedureStepPushSOPClass1          = 380,
00888 UnifiedProcedureStepWatchSOPClass1         = 381,
00889 UnifiedProcedureStepPullSOPClass1          = 382,
00890 UnifiedProcedureStepEventSOPClass1         = 383,
00891 RTBeamsDeliveryInstructionStorage           = 384,
00892 RTConventionalMachineVerification          = 385,
00893 RTIonMachineVerification                    = 386,
00894 RTBrachyApplicationSetupDeliveryInstructionStorage = 387,
00895 HangingProtocolInformationModelGET          = 388,
00896 ColorPaletteStorage                         = 389,
00897 ColorPaletteQueryRetrieveInformationModelFIND = 390,
00898 ColorPaletteQueryRetrieveInformationModelMOVE = 391,
00899 ColorPaletteQueryRetrieveInformationModelGET  = 392,
00900 GenericImplantTemplateStorage                = 393,
00901 GenericImplantTemplateInformationModelFIND    = 394,
00902 GenericImplantTemplateInformationModelMOVE    = 395,
00903 GenericImplantTemplateInformationModelGET      = 396,
00904 ImplantAssemblyTemplateStorage                = 397,
00905 ImplantAssemblyTemplateInformationModelFIND    = 398,
00906 ImplantAssemblyTemplateInformationModelMOVE    = 399,
00907 ImplantAssemblyTemplateInformationModelGET      = 400,
00908 ImplantTemplateGroupStorage                   = 401,
00909 ImplantTemplateGroupInformationModelFIND        = 402,
00910 ImplantTemplateGroupInformationModelMOVE        = 403,
00911 ImplantTemplateGroupInformationModelGET         = 404,
00912 NativeDICOMModel                           = 405,
00913 AbstractMultiDimensionalImageModel            = 406,
00914 DICOMContentMappingResource                  = 407,
00915 EnhancedMRColorImageStorage                  = 408,
00916 UniversalCoordinatedTime                    = 409
00917 //
00918 //
00920 //
00922 //
00923 // Optionally private UIDs
00924 //
00925 #if 0
00926 Private_1_2_840_113619_4_2,
00927 Private_1_2_840_113619_4_3,
00928 Private_1_3_12_2_1107_5_9_1,
00929 Private_1_2_840_113619_4_26,
00930 Private_1_2_840_113619_4_30,
00931 Private_2_16_840_1_113709_1_5_1,
00932 Private_2_16_840_1_113709_1_2_2,
00933 Private_1_2_840_113543_6_6_1_3_10002,
00934 Private_1_2_392_200036_9116_7_8_1_1_1,
00935 Private_1_2_392_200036_9125_1_1_2,
00936 Private_1_2_840_113619_4_27,
00937 Private_1_3_46_670589_11_0_0_12_1,
00938 Private_1_3_46_670589_11_0_0_12_2,
00939 Private_1_3_46_670589_11_0_0_12_4,
00940 Private_1_3_46_670589_2_3_1_1,

```

```

00941 Private_1_3_46_670589_2_4_1_1,
00942 Private_1_3_46_670589_2_5_1_1,
00943 Private_1_3_46_670589_5_0_1,
00944 Private_1_3_46_670589_5_0_1_1,
00945 Private_1_3_46_670589_5_0_10,
00946 Private_1_3_46_670589_5_0_11,
00947 Private_1_3_46_670589_5_0_11_1,
00948 Private_1_3_46_670589_5_0_12,
00949 Private_1_3_46_670589_5_0_13,
00950 Private_1_3_46_670589_5_0_14,
00951 Private_1_3_46_670589_5_0_2,
00952 Private_1_3_46_670589_5_0_2_1,
00953 Private_1_3_46_670589_5_0_3,
00954 Private_1_3_46_670589_5_0_3_1,
00955 Private_1_3_46_670589_5_0_4,
00956 Private_1_3_46_670589_5_0_7,
00957 Private_1_3_46_670589_5_0_8,
00958 Private_1_3_46_670589_5_0_9,
00959 Private_1_2_752_24_3_7_6,
00960 Private_1_2_752_24_3_7_7,
00961 Private_1_2_840_113619_5_2,
00962 Private_1_3_46_670589_33_1_4_1
00963 #endif
00964 //
00965 //
00967
00968 } TSName;
00969
00970
00971 typedef const char* const (*TransferSyntaxStringsType)[2];
00972 static TransferSyntaxStringsType GetTransferSyntaxStrings();
00973 static const char * const *GetTransferSyntaxString(unsigned int ts);
00974 static unsigned int GetNumberOfTransferSyntaxStrings();
00975
00976
00977 // TODO: Because I would like a dual signature for TSType and TSName, C++ won't let me do it...
00978 static const char* GetUIDString(/*TSType*/ unsigned int ts);
00979 static const char* GetUIDName(/*TSType*/ unsigned int ts);
00980
00983 bool SetFromUID(const char *str);
00984
00987 const char *GetName() const;
00988
00991 const char *GetString() const;
00992
00993 operator TSType () const { return TSField; }
00994 UIDs() = default;
00995
00996 private:
00997     TSType TSField;
00998 };
00999 //-----
01000 inline std::ostream &operator<<(std::ostream &_os, const UIDs &uid)
01001 {
01002     _os << uid.GetString() << " -> " << uid.GetName();
01003     return _os;
01004 }
01005 }
01006
01007 } // end namespace gdcm
01008
01009 #endif //GDCMUIDS_H

```

13.109 gdcmAttribute.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"

```


Namespaces

- namespace [gdcm](#)

13.110 gdcmAttribute.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMATATTRIBUTE_H
00015 #define GDCMATATTRIBUTE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmTagToType.h"
00020 #include "gdcmVM.h"
00021 #include "gdcmElement.h"
00022 #include "gdcmDataElement.h"
00023 #include "gdcmDataSet.h"
00024 #include "gdcmStaticAssert.h"
00025
00026 #include <string>
00027 #include <vector>
00028 #include <sstream>
00029
00030 namespace gdcm_ns
00031 {
00032
00033 struct void_;
00034
00035 // Declaration, also serve as forward declaration
00036 template<int T> class VRVLSIZE;
00037
00038 // Implementation when VL is coded on 16 bits:
00039 template<> class VRVLSIZE<0> {
00040 public:
00041     static inline uint16_t Read(std::istream &_is) {
00042         uint16_t l;
00043         _is.read((char*)&l, 2);
00044         return l;
00045     }
00046
00047     static inline void Write(std::ostream &os) { (void)os; }
00048 }
00049 };
00050 // Implementation when VL is coded on 32 bits:
00051 template<> class VRVLSIZE<1> {
00052 public:
00053     static inline uint32_t Read(std::istream &_is) {
00054         char dummy[2];
00055         _is.read(dummy, 2);
00056
00057         uint32_t l;
00058         _is.read((char*)&l, 4);
00059         return l;
00060     }
00061
00062     static inline void Write(std::ostream &os) { (void)os; }
00063 }
00064 };
00065

```

```

00081 template<uint16_t Group, uint16_t Element,
00082     long long TVR = TagToType<Group, Element>::VRType, // can the user override this value ?
00083     int TVM = TagToType<Group, Element>::VMType // can the user override this value ?
00084     /*typename SQAttribute = void_*/ > // if only I had variadic template...
00085 class Attribute
00086 {
00087 public:
00088     typedef typename VRToType<TVR>::Type ArrayType;
00089     enum { VMType = VMToLength<TVM>::Length };
00090     ArrayType Internal[VMToLength<TVM>::Length];
00091
00092     // Make sure that user specified VR/VM are compatible with the public dictionary:
00093     GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
00094     GDCM_STATIC_ASSERT( ((VM::VMType)TVM & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00095     GDCM_STATIC_ASSERT( (((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TVM == VM::VM1) )
00096         || !((VR::VRType)TVR & VR::VR_VM1) );
00097
00098     static Tag GetTag() { return Tag(Group,Element); }
00099     static VR GetVR() { return (VR::VRType)TVR; }
00100     static VM GetVM() { return (VM::VMType)TVM; }
00101
00102     // The following two methods do make sense only in case of public element,
00103     // when the template is intanciated with private element the VR/VM are simply
00104     // defaulted to allow everything (see gdcmtagToType.h default template for TagToType)
00105     static VR GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
00106     static VM GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
00107
00108     // Some extra dummy checks:
00109     // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
00110
00111     unsigned int GetNumberOfValues() const {
00112         return VMToLength<TVM>::Length;
00113     }
00114     // Implementation of Print is common to all Mode (ASCII/Binary)
00115     // TODO: Can we print a \ when in ASCII...well I don't think so
00116     // it would mean we used a bad VM then, right ?
00117     void Print(std::ostream &os) const {
00118         os << GetTag() << " ";
00119         os << TagToType<Group,Element>::GetVRString() << " ";
00120         os << TagToType<Group,Element>::GetVMString() << " ";
00121         os << Internal[0]; // VM is at least guarantee to be one
00122         for(unsigned int i=1; i<GetNumberOfValues(); ++i)
00123             os << "," << Internal[i];
00124     }
00125
00126     // copy:
00127     //ArrayType GetValue(unsigned int idx = 0) {
00128     //    gdcmt_assert( idx < GetNumberOfValues() );
00129     //    return Internal[idx];
00130     //}
00131     //ArrayType operator[] (unsigned int idx) {
00132     //    return GetValue(idx);
00133     //}
00134     // FIXME: is this always a good idea ?
00135     // I do not think so, I prefer operator
00136     //operator ArrayType () const { return Internal[0]; }
00137
00138     bool operator==(const Attribute &att) const
00139     {
00140         return std::equal(Internal, Internal+GetNumberOfValues(),
00141             att.GetValues());
00142     }
00143     bool operator!=(const Attribute &att) const
00144     {
00145         return !std::equal(Internal, Internal+GetNumberOfValues(),
00146             att.GetValues());
00147     }
00148     bool operator<(const Attribute &att) const
00149     {
00150         return std::lexicographical_compare(Internal, Internal+GetNumberOfValues(),
00151             att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
00152     }
00153
00154     ArrayType &GetValue(unsigned int idx = 0) {
00155         gdcmt_assert( idx < GetNumberOfValues() );
00156         return Internal[idx];
00157     }
00158     ArrayType & operator[] (unsigned int idx) {
00159         return GetValue(idx);
00160     }
00161     // const reference

```

```

00162 ArrayType const &GetValue(unsigned int idx = 0) const {
00163     gdcml_assert( idx < GetNumberOfValues() );
00164     return Internal[idx];
00165 }
00166 ArrayType const & operator[] (unsigned int idx) const {
00167     return GetValue(idx);
00168 }
00169 void SetValue(ArrayType v, unsigned int idx = 0) {
00170     gdcml_assert( idx < GetNumberOfValues() );
00171     Internal[idx] = v;
00172 }
00173 void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
00174     gdcml_assert( array && numel && numel == GetNumberOfValues() );
00175     // std::copy is smarter than a memcpy, and will call memcpy when POD type
00176     std::copy(array, array+numel, Internal);
00177 }
00178 const ArrayType* GetValues() const {
00179     return Internal;
00180 }
00181
00182 // API to talk to the run-time layer: gdcm::DataElement
00183 DataElement GetAsDataElement() const {
00184     DataElement ret( GetTag() );
00185     std::ostringstream os;
00186     // os.imbue(std::locale::classic()); // This is not required AFAIK
00187     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00188         GetNumberOfValues(),os);
00189     ret.SetVR( GetVR() );
00190     gdcml_assert( ret.GetVR() != VR::SQ );
00191     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00192     {
00193         if( GetVR() != VR::UI )
00194         {
00195             if( os.str().size() % 2 )
00196             {
00197                 os << " ";
00198             }
00199         }
00200     }
00201     VL::Type osStrSize = (VL::Type)os.str().size();
00202     ret.SetByteValue( os.str().c_str(), osStrSize );
00203     return ret;
00204 }
00205
00206 void SetFromDataElement(DataElement const &de) {
00207     // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
00208     gdcml_assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
00209     gdcml_assert( GetVR() != VR::INVALID );
00210     gdcml_assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID cannot
00211     use the & operator
00212     if( de.IsEmpty() ) return;
00213     const ByteValue *bv = de.GetByteValue();
00214     #ifdef GDCM_WORDS_BIGENDIAN
00215     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00216     #else
00217     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00218     #endif
00219     {
00220         SetByteValue(bv);
00221     }
00222     else
00223     {
00224         SetByteValueNoSwap(bv);
00225     }
00226 }
00227 void Set(DataSet const &ds) {
00228     SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00229 }
00230 void SetFromDataSet(DataSet const &ds) {
00231     if( ds.FindDataElement( Tag(Group,Element) ) &&
00232         !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
00233     {
00234         SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00235     }
00236 }
00237 protected:
00238 void SetByteValueNoSwap(const ByteValue *bv) {
00239     if( !bv ) return; // That would be bad...
00240     gdcml_assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00241     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00242     // {

```

```

00242 // // always do a copy !
00243 // SetValues(bv->GetPointer(), bv->GetLength());
00244 // }
00245 //else
00246 {
00247     std::stringstream ss;
00248     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00249     ss.str( s );
00250     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00251         GetNumberOfValues(),ss);
00252 }
00253 }
00254 void SetByteValue(const ByteValue *bv) {
00255     if( !bv ) return; // That would be bad...
00256     gdcml_assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00257     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00258     // {
00259     // // always do a copy !
00260     // SetValues(bv->GetPointer(), bv->GetLength());
00261     // }
00262     //else
00263     {
00264         std::stringstream ss;
00265         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00266         ss.str( s );
00267         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00268             GetNumberOfValues(),ss);
00269     }
00270 }
00271 #if 0 // TODO FIXME the implicit way:
00272 // explicit:
00273 void Read(std::istream &_is) {
00274     const uint16_t cref[] = { Group, Element };
00275     uint16_t c[2];
00276     _is.read((char*)&c, sizeof(c));
00277     gdcml_assert( c[0] == cref[0] && c[1] == cref[1] );
00278     char vr[2];
00279     _is.read(vr, 2); // Check consistency ?
00280     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00281     uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
00282     l /= sizeof( typename VRToType<TVR>::Type );
00283     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00284         l,_is);
00285 }
00286 void Write(std::ostream &_os) const {
00287     uint16_t c[] = { Group, Element };
00288     _os.write((char*)&c, 4);
00289     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00290     _os.write((char*)&l, 4);
00291     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00292         GetLength(),_os);
00293 }
00294 void Read(std::istream &_is) {
00295     uint16_t cref[] = { Group, Element };
00296     uint16_t c[2];
00297     _is.read((char*)&c, 4);
00298     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00299     uint32_t l;
00300     _is.read((char*)&l, 4);
00301     l /= sizeof( typename VRToType<TVR>::Type );
00302     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00303         l,_is);
00304 }
00305 void Write(std::ostream &_os) const {
00306     uint16_t c[] = { Group, Element };
00307     _os.write((char*)&c, 4);
00308     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00309     _os.write((char*)&l, 4);
00310     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00311         GetLength(),_os);
00312 }
00313 #endif
00314 };
00315 };
00316
00317 template<uint16_t Group, uint16_t Element, long long TVR >
00318 class Attribute<Group,Element,TVR,VM::VM1>
00319 {
00320 public:
00321     typedef typename VRToType<TVR>::Type ArrayType;
00322     enum { VMType = VMToLength<VM::VM1>::Length };

```

```

00323 //ArrayType Internal[VMToLength<TVM>::Length];
00324 ArrayType Internal;
00325 GDCM_STATIC_ASSERT( VMToLength<VM::VM1>::Length == 1 );
00326
00327 // Make sure that user specified VR/VM are compatible with the public dictionary:
00328 GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
00329 GDCM_STATIC_ASSERT( ((VM::VMType)VM::VM1 & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00330 GDCM_STATIC_ASSERT( (((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)VM::VM1 == VM::VM1) )
00331 || !((VR::VRType)TVR & VR::VR_VM1) );
00332
00333 static Tag GetTag() { return Tag(Group,Element); }
00334 static VR GetVR() { return (VR::VRType)TVR; }
00335 static VM GetVM() { return (VM::VMType)VM::VM1; }
00336
00337 // The following two methods do make sense only in case of public element,
00338 // when the template is intanciated with private element the VR/VM are simply
00339 // defaulted to allow everything (see gdcMTagToType.h default template for TagToType)
00340 static VR GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
00341 static VM GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }
00342
00343 // Some extra dummy checks:
00344 // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.
00345
00346 unsigned int GetNumberOfValues() const {
00347     return VMToLength<VM::VM1>::Length;
00348 }
00349 // Implementation of Print is common to all Mode (ASCII/Binary)
00350 // TODO: Can we print a \ when in ASCII...well I don't think so
00351 // it would mean we used a bad VM then, right ?
00352 void Print(std::ostream &os) const {
00353     os << GetTag() << " ";
00354     os << TagToType<Group,Element>::GetVRString() << " ";
00355     os << TagToType<Group,Element>::GetVMString() << " ";
00356     os << Internal; // VM is at least guarantee to be one
00357 }
00358 // copy:
00359 //ArrayType GetValue(unsigned int idx = 0) {
00360 //    gdcM_assert( idx < GetNumberOfValues() );
00361 //    return Internal[idx];
00362 //}
00363 //ArrayType operator[] (unsigned int idx) {
00364 //    return GetValue(idx);
00365 //}
00366 // FIXME: is this always a good idea ?
00367 // I do not think so, I prefer operator
00368 //operator ArrayType () const { return Internal[0]; }
00369
00370 bool operator==(const Attribute &att) const
00371 {
00372     return std::equal(&Internal, &Internal+GetNumberOfValues(),
00373         att.GetValues());
00374 }
00375 bool operator!=(const Attribute &att) const
00376 {
00377     return !std::equal(&Internal, &Internal+GetNumberOfValues(),
00378         att.GetValues());
00379 }
00380 bool operator<(const Attribute &att) const
00381 {
00382     return std::lexicographical_compare(&Internal, &Internal+GetNumberOfValues(),
00383         att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
00384 }
00385
00386 ArrayType &GetValue() {
00387     // gdcM_assert( idx < GetNumberOfValues() );
00388     return Internal;
00389 }
00390 // ArrayType & operator[] (unsigned int idx) {
00391 //     return GetValue(idx);
00392 // }
00393 // const reference
00394 ArrayType const &GetValue() const {
00395     //gdcM_assert( idx < GetNumberOfValues() );
00396     return Internal;
00397 }
00398 //ArrayType const & operator[] () const {
00399 //     return GetValue();
00400 //}
00401 void SetValue(ArrayType v) {
00402     // gdcM_assert( idx < GetNumberOfValues() );
00403     Internal = v;

```

```

00404 }
00405 /* void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
00406     gdcmm_assert( array && numel && numel == GetNumberOfValues() );
00407     // std::copy is smarter than a memcpy, and will call memcpy when POD type
00408     std::copy(array, array+numel, Internal);
00409 }
00410 */
00411
00412 // FIXME Should we remove this function ?
00413 const ArrayType* GetValues() const {
00414     return &Internal;
00415 }
00416
00417 // API to talk to the run-time layer: gdcmm::DataElement
00418 DataElement GetAsDataElement() const {
00419     DataElement ret( Tag(Group,Element) );
00420     std::ostream os;
00421     // os.imbue(std::locale::classic()); // This is not required AFAIK
00422     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(&Internal,
00423         GetNumberOfValues(),os);
00424     ret.SetVR( GetVR() );
00425     gdcmm_assert( ret.GetVR() != VR::SQ );
00426     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00427     {
00428         if( GetVR() != VR::UI )
00429         {
00430             if( os.str().size() % 2 )
00431             {
00432                 os << " ";
00433             }
00434         }
00435     }
00436     VL::Type osStrSize = (VL::Type)os.str().size();
00437     ret.SetByteValue( os.str().c_str(), osStrSize );
00438     return ret;
00439 }
00440
00441 void SetFromDataElement(DataElement const &de) {
00442     // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
00443     gdcmm_assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
00444     gdcmm_assert( GetVR() != VR::INVALID );
00445     gdcmm_assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID cannot
    use the & operator
00446     if( de.IsEmpty() ) return;
00447     const ByteValue *bv = de.GetByteValue();
00448 #ifdef GDCM_WORDS_BIGENDIAN
00449     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00450     #else
00451     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00452     #endif
00453     {
00454         SetByteValue(bv);
00455     }
00456     else
00457     {
00458         SetByteValueNoSwap(bv);
00459     }
00460 }
00461 void Set(DataSet const &ds) {
00462     SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00463 }
00464 void SetFromDataSet(DataSet const &ds) {
00465     if( ds.FindDataElement( Tag(Group,Element) ) &&
00466         !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
00467     {
00468         SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
00469     }
00470 }
00471 protected:
00472 void SetByteValueNoSwap(const ByteValue *bv) {
00473     if( !bv ) return; // That would be bad...
00474     gdcmm_assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00475     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00476     // {
00477     //     // always do a copy !
00478     //     SetValues(bv->GetPointer(), bv->GetLength());
00479     // }
00480     //else
00481     {
00482         std::stringstream ss;
00483         std::string s = std::string( bv->GetPointer(), bv->GetLength() );

```

```

00484     ss.str( s );
00485     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(&Internal,
00486         GetNumberOfValues(),ss);
00487     }
00488 }
00489 void SetByteValue(const ByteValue *bv) {
00490     if( !bv ) return; // That would be bad...
00491     gdcm_assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
00492     //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
00493     // {
00494     //     // always do a copy !
00495     //     SetValues(bv->GetPointer(), bv->GetLength());
00496     // }
00497     //else
00498     {
00499         std::stringstream ss;
00500         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00501         ss.str( s );
00502         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(&Internal,
00503             GetNumberOfValues(),ss);
00504     }
00505 }
00506 #if 0 // TODO FIXME the implicit way:
00507 // explicit:
00508 void Read(std::istream &_is) {
00509     const uint16_t cref[] = { Group, Element };
00510     uint16_t c[2];
00511     _is.read((char*)&c, sizeof(c));
00512     gdcm_assert( c[0] == cref[0] && c[1] == cref[1] );
00513     char vr[2];
00514     _is.read(vr, 2); // Check consistency ?
00515     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00516     uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
00517     l /= sizeof( typename VRToType<TVR>::Type );
00518     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00519         l,_is);
00520 }
00521 void Write(std::ostream &_os) const {
00522     uint16_t c[] = { Group, Element };
00523     _os.write((char*)&c, 4);
00524     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00525     _os.write((char*)&l, 4);
00526     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00527         GetLength(),_os);
00528 }
00529 void Read(std::istream &_is) {
00530     uint16_t cref[] = { Group, Element };
00531     uint16_t c[2];
00532     _is.read((char*)&c, 4);
00533     const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
00534     uint32_t l;
00535     _is.read((char*)&l, 4);
00536     l /= sizeof( typename VRToType<TVR>::Type );
00537     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00538         l,_is);
00539 }
00540 void Write(std::ostream &_os) const {
00541     uint16_t c[] = { Group, Element };
00542     _os.write((char*)&c, 4);
00543     uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
00544     _os.write((char*)&l, 4);
00545     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00546         GetLength(),_os);
00547 }
00548 #endif
00549
00550 };
00551
00552 // No need to repeat default template arg, since primary template
00553 // will be used to generate the default arguments
00554 template<uint16_t Group, uint16_t Element, long long TVR >
00555 class Attribute<Group,Element,TVR,VM::VM1_n>
00556 {
00557 public:
00558     typedef typename VRToType<TVR>::Type ArrayType;
00559
00560     // Make sure that user specified VR/VM are compatible with the public dictionary:
00561     GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
00562     GDCM_STATIC_ASSERT( (VM::VM1_n & (VM::VMType)(TagToType<Group, Element>::VMType)) );
00563     GDCM_STATIC_ASSERT( (((VR::VRType)TVR & VR::VR_VM1) &&
        ((VM::VMType)TagToType<Group,Element>::VMType == VM::VM1) )

```



```

00564         || !((VR::VRType)TVR & VR::VR_VM1) ) );
00565
00566 static Tag GetTag() { return Tag(Group,Element); }
00567 static VR GetVR() { return (VR::VRType)TVR; }
00568 static VM GetVM() { return VM::VM1_n; }
00569
00570 static VR GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
00571 static VM GetDictVM() { return GetVM(); }
00572
00573 // This the way to prevent default initialization
00574 explicit Attribute() { Internal=nullptr; Length=0; Own = true; }
00575 ~Attribute() {
00576     if( Own ) {
00577         delete[] Internal;
00578     }
00579     Internal = nullptr; // paranoid
00580 }
00581
00582 unsigned int GetNumberOfValues() const { return Length; }
00583
00584 void SetNumberOfValues(unsigned int numel)
00585 {
00586     SetValues(nullptr, numel, true);
00587 }
00588
00589 const ArrayType* GetValues() const {
00590     return Internal;
00591 }
00592 void Print(std::ostream &os) const {
00593     os << GetTag() << " ";
00594     os << GetVR() << " ";
00595     os << GetVM() << " ";
00596     os << Internal[0]; // VM is at least guarantee to be one
00597     for(unsigned int i=1; i<GetNumberOfValues(); ++i)
00598         os << ", " << Internal[i];
00599 }
00600 ArrayType &GetValue(unsigned int idx = 0) {
00601     gdcmm_assert( idx < GetNumberOfValues() );
00602     return Internal[idx];
00603 }
00604 ArrayType &operator[] (unsigned int idx) {
00605     return GetValue(idx);
00606 }
00607 // const reference
00608 ArrayType const &GetValue(unsigned int idx = 0) const {
00609     gdcmm_assert( idx < GetNumberOfValues() );
00610     return Internal[idx];
00611 }
00612 ArrayType const & operator[] (unsigned int idx) const {
00613     return GetValue(idx);
00614 }
00615 void SetValue(unsigned int idx, ArrayType v) {
00616     gdcmm_assert( idx < GetNumberOfValues() );
00617     Internal[idx] = v;
00618 }
00619 void SetValue(ArrayType v) { SetValue(0, v); }
00620
00621 void SetValues(const ArrayType *array, unsigned int numel, bool own = false)
00622 {
00623     if( Internal ) // were we used before ?
00624     {
00625         // yes !
00626         if( Own ) delete[] Internal;
00627         Internal = nullptr;
00628     }
00629     Own = own;
00630     Length = numel;
00631     gdcmm_assert( Internal == nullptr );
00632     if( own ) // make a copy:
00633     {
00634         Internal = new ArrayType[numel];
00635         if( array && numel )
00636             std::copy(array, array+numel, Internal);
00637     }
00638     else // pass pointer
00639     {
00640         Internal = const_cast<ArrayType*>(array);
00641     }
00642     // postcondition
00643     gdcmm_assert( numel == GetNumberOfValues() );
00644 }

```



```

00645
00646 DataElement GetAsDataElement() const {
00647     DataElement ret( GetTag() );
00648     std::ostringstream os;
00649     if( Internal )
00650     {
00651         EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00652             GetNumberOfValues(),os);
00653         if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00654         {
00655             if( GetVR() != VR::UI )
00656             {
00657                 if( os.str().size() % 2 )
00658                 {
00659                     os << " ";
00660                 }
00661             }
00662         }
00663     }
00664     ret.SetVR( GetVR() );
00665     gdcM_assert( ret.GetVR() != VR::SQ );
00666     VL::Type osStrSize = (VL::Type) os.str().size();
00667     ret.SetByteValue( os.str().c_str(), osStrSize);
00668     return ret;
00669 }
00670 void SetFromDataElement(DataElement const &de) {
00671     // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
00672     gdcM_assert( GetTag() == de.GetTag() || GetTag().GetGroup() == 0x6000
00673         || GetTag().GetGroup() == 0x5000 );
00674     gdcM_assert( GetVR().Compatible( de.GetVR() ) ); // In case of VR::INVALID cannot use the & operator
00675     gdcM_assert( !de.IsEmpty() );
00676     const ByteValue *bv = de.GetByteValue();
00677     SetByteValue(bv);
00678 }
00679 void Set(DataSet const &ds) {
00680     SetFromDataElement( ds.GetDataElement( GetTag() ) );
00681 }
00682 void SetFromDataSet(DataSet const &ds) {
00683     if( ds.FindDataElement( GetTag() ) &&
00684         !ds.GetDataElement( GetTag() ).IsEmpty() )
00685     {
00686         SetFromDataElement( ds.GetDataElement( GetTag() ) );
00687     }
00688 }
00689 protected:
00690 void SetByteValue(const ByteValue *bv) {
00691     gdcM_assert( bv ); // FIXME
00692     std::stringstream ss;
00693     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00694     Length = bv->GetLength(); // HACK FIXME
00695     ss.str( s );
00696     ArrayType *internal;
00697     ArrayType buffer[256];
00698     if( bv->GetLength() < 256 )
00699     {
00700         internal = buffer;
00701     }
00702     else
00703     {
00704         internal = new ArrayType[(VL::Type)bv->GetLength()]; // over allocation
00705     }
00706     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(internal, Length, ss);
00707     SetValues( internal, Length, true );
00708     if( !(bv->GetLength() < 256) )
00709     {
00710         delete[] internal;
00711     }
00712     //EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00713     //    GetNumberOfValues(),ss);
00714 }
00715
00716 private:
00717     ArrayType *Internal;
00718     unsigned int Length;
00719     bool Own : 1;
00720 };
00721
00722 template<uint16_t Group, uint16_t Element, long long TVR>
00723 class Attribute<Group,Element,TVR,VM::VM1_3> : public Attribute<Group,Element,TVR,VM::VM1_n>
00724 {
00725 public:

```

```

00726 VM GetVM() const { return VM::VM1_3; }
00727 };
00728
00729 template<uint16_t Group, uint16_t Element, long long TVR>
00730 class Attribute<Group,Element,TVR,VM::VM1_8> : public Attribute<Group,Element,TVR,VM::VM1_n>
00731 {
00732 public:
00733 VM GetVM() const { return VM::VM1_8; }
00734 };
00735
00736 template<uint16_t Group, uint16_t Element, long long TVR>
00737 class Attribute<Group,Element,TVR,VM::VM2_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
00738 {
00739 public:
00740 VM GetVM() const { return VM::VM2_n; }
00741 };
00742
00743 template<uint16_t Group, uint16_t Element, long long TVR>
00744 class Attribute<Group,Element,TVR,VM::VM2_2n> : public Attribute<Group,Element,TVR,VM::VM2_n>
00745 {
00746 public:
00747 static VM GetVM() { return VM::VM2_2n; }
00748 };
00749
00750 template<uint16_t Group, uint16_t Element, long long TVR>
00751 class Attribute<Group,Element,TVR,VM::VM3_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
00752 {
00753 public:
00754 static VM GetVM() { return VM::VM3_n; }
00755 };
00756
00757 template<uint16_t Group, uint16_t Element, long long TVR>
00758 class Attribute<Group,Element,TVR,VM::VM3_3n> : public Attribute<Group,Element,TVR,VM::VM3_n>
00759 {
00760 public:
00761 static VM GetVM() { return VM::VM3_3n; }
00762 };
00763
00764
00765 // For particular case for ASCII string
00766 // WARNING: This template explicitly instantiates a particular
00767 // EncodingImplementation THEREFORE it is required to be declared after the
00768 // EncodingImplementation is needs (doh!)
00769 #if 0
00770 template<int TVM>
00771 class Attribute<TVM>
00772 {
00773 public:
00774 Attribute(const char array[])
00775 {
00776 unsigned int i = 0;
00777 const char sep = '\\';
00778 std::string sarray = array;
00779 std::string::size_type pos1 = 0;
00780 std::string::size_type pos2 = sarray.find(sep, pos1+1);
00781 while(pos2 != std::string::npos)
00782 {
00783 Internal[i++] = sarray.substr(pos1, pos2-pos1);
00784 pos1 = pos2+1;
00785 pos2 = sarray.find(sep, pos1+1);
00786 }
00787 Internal[i] = sarray.substr(pos1, pos2-pos1);
00788 // Shouldn't we do the contrary, since we know how many separators
00789 // (and default behavior is to discard anything after the VM declared
00790 gdcml_assert( GetLength()-1 == i );
00791 }
00792
00793 unsigned long GetLength() const {
00794 return VMToLength<TVM>::Length;
00795 }
00796 // Implementation of Print is common to all Mode (ASCII/Binary)
00797 void Print(std::ostream &_os) const {
00798 _os << Internal[0]; // VM is at least guarantee to be one
00799 for(int i=1; i<VMToLength<TVM>::Length; ++i)
00800 _os << " " << Internal[i];
00801 }
00802
00803 void Read(std::istream &_is) {
00804 EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(),_is);
00805 }
00806 void Write(std::ostream &_os) const {

```

```

00807   EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(),_os);
00808   }
00809 private:
00810   typename String Internal[VMToLength<TVM>::Length];
00811 };
00812
00813 template< int TVM>
00814 class Attribute<VR::PN, TVM> : public StringAttribute<TVM>
00815 {
00816 };
00817 #endif
00818
00819 #if 0
00820
00821 // Implementation for the undefined length (dynamically allocated array)
00822 template<int TVR>
00823 class Attribute<TVR, VM::VM1_n>
00824 {
00825 public:
00826   // This the way to prevent default initialization
00827   explicit Attribute() { Internal=0; Length=0; }
00828   ~Attribute() {
00829     delete[] Internal;
00830     Internal = 0;
00831   }
00832
00833   // Length manipulation
00834   // SetLength should really be protected anyway...all operation
00835   // should go through SetArray
00836   unsigned long GetLength() const { return Length; }
00837   typedef typename VRToType<TVR>::Type ArrayType;
00838   void SetLength(unsigned long len) {
00839     const unsigned int size = sizeof(ArrayType);
00840     if( len ) {
00841       if( len > Length ) {
00842         // perform realloc
00843         gdcm_assert( (len / size) * size == len );
00844         ArrayType *internal = new ArrayType[len / size];
00845         memcpy(internal, Internal, Length * size);
00846         delete[] Internal;
00847         Internal = internal;
00848       }
00849     }
00850     Length = len / size;
00851   }
00852
00853   // If save is set to zero user should not delete the pointer
00854   //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
00855   void SetArray(const ArrayType *array, unsigned long len,
00856     bool save = false) {
00857     if( save ) {
00858       SetLength(len); // realloc
00859       memcpy(Internal, array, len/*sizeof(ArrayType)**/);
00860     }
00861     else {
00862       // TODO rewrite this stupid code:
00863       Length = len;
00864       //Internal = array;
00865       gdcm_assert(0);
00866     }
00867   }
00868   // Implementation of Print is common to all Mode (ASCII/Binary)
00869   void Print(std::ostream &_os) const {
00870     gdcm_assert( Length );
00871     gdcm_assert( Internal );
00872     _os << Internal[0]; // VM is at least guarantee to be one
00873     const unsigned long length = GetLength() < 25 ? GetLength() : 25;
00874     for(unsigned long i=1; i<length; ++i)
00875       _os << " " << Internal[i];
00876   }
00877   void Read(std::istream &_is) {
00878     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00879       GetLength(),_is);
00880   }
00881   void Write(std::ostream &_os) const {
00882     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00883       GetLength(),_os);
00884   }
00885
00886   Attribute(const Attribute&_val) {
00887     if( this != &_amp;_val ) {

```

```

00888     *this = __val;
00889     }
00890     }
00891
00892     Attribute &operator=(const Attribute &__val) {
00893         Length = 0; // SYITF
00894         Internal = 0;
00895         SetArray(__val.Internal, __val.Length, true);
00896         return *this;
00897     }
00898
00899 private:
00900     typename VRToType<TVR>::Type *Internal;
00901     unsigned long Length; // unsigned int ??
00902 };
00903
00904 //template <int TVM = VM::VM1_n>
00905 //class Attribute<VR::OB, TVM > : public Attribute<VR::OB, VM::VM1_n> {};
00906
00907 // Partial specialization for derivatives of 1-n : 2-n, 3-n ...
00908 template<int TVR>
00909 class Attribute<TVR, VM::VM2_n> : public Attribute<TVR, VM::VM1_n>
00910 {
00911 public:
00912     typedef Attribute<TVR, VM::VM1_n> Parent;
00913     void SetLength(int len) {
00914         if( len <= 1 ) return;
00915         Parent::SetLength(len);
00916     }
00917 };
00918 template<int TVR>
00919 class Attribute<TVR, VM::VM2_2n> : public Attribute<TVR, VM::VM2_n>
00920 {
00921 public:
00922     typedef Attribute<TVR, VM::VM2_n> Parent;
00923     void SetLength(int len) {
00924         if( len % 2 ) return;
00925         Parent::SetLength(len);
00926     }
00927 };
00928 template<int TVR>
00929 class Attribute<TVR, VM::VM3_n> : public Attribute<TVR, VM::VM1_n>
00930 {
00931 public:
00932     typedef Attribute<TVR, VM::VM1_n> Parent;
00933     void SetLength(int len) {
00934         if( len <= 2 ) return;
00935         Parent::SetLength(len);
00936     }
00937 };
00938 template<int TVR>
00939 class Attribute<TVR, VM::VM3_3n> : public Attribute<TVR, VM::VM3_n>
00940 {
00941 public:
00942     typedef Attribute<TVR, VM::VM3_n> Parent;
00943     void SetLength(int len) {
00944         if( len % 3 ) return;
00945         Parent::SetLength(len);
00946     }
00947 };
00948
00949
00950 //template<int T> struct VRToLength;
00951 //template <> struct VRToLength<VR::AS>
00952 //{ enum { Length = VM::VM1 }; }
00953 //template<>
00954 //class Attribute<VR::AS> : public Attribute<VR::AS, VRToLength<VR::AS>::Length >
00955
00956 // only 0010 1010 AS 1 Patient's Age
00957 template<>
00958 class Attribute<VR::AS, VM::VM5>
00959 {
00960 public:
00961     char Internal[VMToLength<VM::VM5>::Length];
00962     void Print(std::ostream &__os) const {
00963         __os << Internal;
00964     }
00965 };
00966
00967 template <>
00968 class Attribute<VR::OB, VM::VM1> : public Attribute<VR::OB, VM::VM1_n> {};

```

```

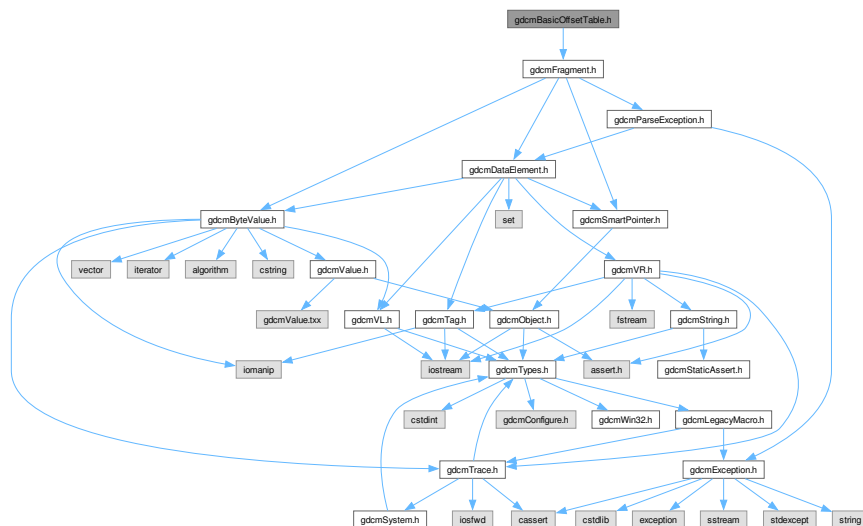
00969 // Make it impossible to compile any other cases:
00970 template <int TVM> class Attribute<VR::OB, TVM>;
00971
00972 // Same for OW:
00973 template <>
00974 class Attribute<VR::OW, VM::VM1> : public Attribute<VR::OW, VM::VM1_n> {};
00975 // Make it impossible to compile any other cases:
00976 template <int TVM> class Attribute<VR::OW, TVM>;
00977 #endif
00978
00979 #if 0
00980 template<>
00981 class Attribute<0x7fe0,0x0010, VR::OW, VM::VM1>
00982 {
00983 public:
00984   char *Internal;
00985   unsigned long Length; // unsigned int ??
00986
00987   void Print(std::ostream &_os) const {
00988     _os << Internal[0];
00989   }
00990   void SetBytes(char *bytes, unsigned long length) {
00991     Internal = bytes;
00992     Length = length;
00993   }
00994   void Read(std::istream &_is) {
00995     uint16_t c[2];
00996     _is.read((char*)&c, 4);
00997     uint32_t l;
00998     _is.read((char*)&l, 4);
00999     Length = l;
01000     _is.read( Internal, Length );
01001   }
01002   void Write(std::ostream &_os) const {
01003     uint16_t c[] = {0x7fe0, 0x0010};
01004     _os.write((char*)&c, 4);
01005     _os.write((char*)&Length, 4);
01006     _os.write( Internal, Length );
01007   }
01008 };
01009 #endif
01010
01011 /*
01012 // Removing Attribute for SQ for now...
01013 template<uint16_t Group, uint16_t Element, typename SQA>
01014 class Attribute<Group,Element, VR::SQ, VM::VM1, SQA>
01015 {
01016 public:
01017   SQA sqa;
01018   void Print(std::ostream &_os) const {
01019     _os << Tag(Group,Element);
01020     sqa.Print(_os << std::endl << '\t');
01021   }
01022   void Write(std::ostream &_os) const {
01023     uint16_t c[] = {Group, Element};
01024     _os.write((char*)&c, 4);
01025     uint32_t undef = 0xffffffff;
01026     _os.write((char*)&undef, 4);
01027     uint16_t item_beg[] = {0xfffe,0xe000};
01028     _os.write((char*)&item_beg, 4);
01029     _os.write((char*)&undef, 4);
01030     sqa.Write(_os);
01031     uint16_t item_end[] = {0xfffe,0xe00d};
01032     _os.write((char*)&item_end, 4);
01033     uint32_t zero = 0x0;
01034     _os.write((char*)&zero, 4);
01035     uint16_t seq_end[] = {0xfffe, 0xe0dd};
01036     _os.write((char*)&seq_end, 4);
01037     _os.write((char*)&zero, 4);
01038   }
01039 };
01040 */
01041
01042 } // namespace gdcm_ns
01043
01044 #endif //GDCMATATTRIBUTE_H

```

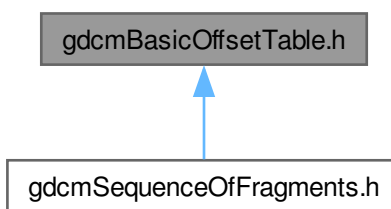
13.111 gdcmBasicOffsetTable.h File Reference

#include "gdcmFragment.h"

Include dependency graph for gdcmBasicOffsetTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::BasicOffsetTable](#)
Class to represent a [BasicOffsetTable](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)`

13.112 gdcmBasicOffsetTable.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  #ifndef GDCMBASICOFFSETTABLE_H
00016  #define GDCMBASICOFFSETTABLE_H
00017
00018  #include "gdcmFragment.h"
00019
00020  namespace gdcm_ns
00021  {
00022
00023  class GDCM_EXPORT BasicOffsetTable : public Fragment
00024  {
00025  protected:
00026  // void SetTag(const Tag &t);
00027  public:
00028  BasicOffsetTable() : Fragment() {}
00029  friend std::ostream &operator<<(std::ostream &os, const BasicOffsetTable &val);
00030
00031  /*
00032  VL GetLength() const {
00033  gdcm_assert( !ValueLengthField.IsUndefined() );
00034  gdcm_assert( !ValueField || ValueField->GetLength() == ValueLengthField );
00035  return TagField.GetLength() + ValueLengthField.GetLength()
00036  + ValueLengthField;
00037  }
00038  */
00039
00040  template <typename TSwap>
00041  std::istream &Read(std::istream &is) {
00042  // Superclass
00043  const Tag itemStart(0xffff, 0xe000);
00044  if( !TagField.Read<TSwap>(is) )
00045  {
00046  gdcm_assert(0 && "Should not happen");
00047  return is;
00048  }
00049  //gdcm_assert( TagField == itemStart );
00050  if( TagField != itemStart )
00051  {
00052  // Bug_Siemens_PrivateIconNoItem.dcm
00053  //gdcmDebugMacro( "Could be Bug_Siemens_PrivateIconNoItem.dcm" );
00054  ParseException pe;
00055  pe.SetLastElement(*this);
00056  //throw "SIEMENS Icon thingy";
00057  throw pe;
00058  }
00059  if( !ValueLengthField.Read<TSwap>(is) )
00060  {
00061  gdcm_assert(0 && "Should not happen");
00062  return is;
00063  }
00064  // Self
00065  SmartPointer<ByteValue> bv = new ByteValue;

```

```

00069     bv->SetLength(ValueLengthField);
00070     if( !bv->Read<TSwap>(is) )
00071     {
00072         gdcmaAssertAlwaysMacro(0 && "Should not happen");
00073         return is;
00074     }
00075     ValueField = bv;
00076     return is;
00077 }
00078
00079 /*
00080 template <typename TSwap>
00081 std::ostream &Write(std::ostream &os) const {
00082     const Tag itemStart(0xfffe, 0xe000);
00083     const Tag seqDelItem(0xfffe, 0xe0dd);
00084     if( !TagField.Write<TSwap>(os) )
00085     {
00086         gdcma_assert(0 && "Should not happen");
00087         return os;
00088     }
00089     gdcma_assert( TagField == itemStart );
00090     if( !ValueLengthField.Write<TSwap>(os) )
00091     {
00092         gdcma_assert(0 && "Should not happen");
00093         return os;
00094     }
00095     if( ValueLengthField )
00096     {
00097         // Self
00098         const ByteValue *bv = GetByteValue();
00099         gdcma_assert( bv );
00100         gdcma_assert( bv->GetLength() == ValueLengthField );
00101         if( !bv->Write<TSwap>(os) )
00102         {
00103             gdcma_assert(0 && "Should not happen");
00104             return os;
00105         }
00106     }
00107     return os;
00108 }
00109 */
00110 };
00111 //-----
00112 inline std::ostream &operator<<(std::ostream &os, const BasicOffsetTable &val)
00113 {
00114     os << " BasicOffsetTable Length=" << val.ValueLengthField << std::endl;
00115     if( val.ValueField )
00116     {
00117         const ByteValue *bv = val.GetByteValue();
00118         gdcma_assert( bv );
00119         os << *bv;
00120     }
00121
00122     return os;
00123 }
00124
00125
00126 } // end namespace gdcma_ns
00127
00128 #endif //GDCMBASICOFFSETTABLE_H

```

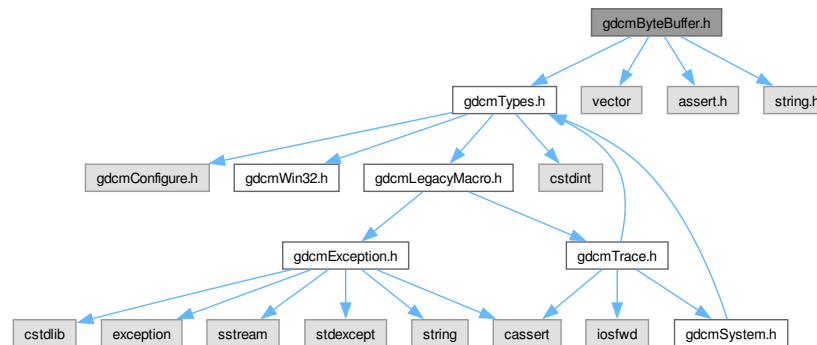
13.113 gdcmaByteBuffer.h File Reference

```

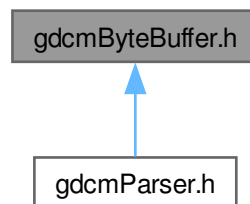
#include "gdcmaTypes.h"
#include <vector>
#include <assert.h>
#include <string.h>

```


Include dependency graph for gdcmByteBuffer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::ByteBuffer`
`ByteBuffer`.

Namespaces

- namespace `gdcm`

13.114 gdcmByteBuffer.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMBYTEBUFFER_H
00015 #define GDCMBYTEBUFFER_H
00016
00017 #include "gdcmTypes.h"
00018 #include <vector>
00019 #include <assert.h>
00020 #include <string.h> // memmove
00021
00022 #error should not be used
00023
00024 namespace gdcm
00025 {
00026     class ByteBuffer
00027     {
00028     public:
00029         static const int InitBufferSize = 1024;
00030         ByteBuffer() : Start(0), End(0), Limit(0) {}
00031         char *Get(int len)
00032         {
00033             char *buffer = &Internal[0];
00034             if (len > Limit - End)
00035             {
00036                 // FIXME avoid integer overflow
00037                 int neededSize = len + (End - Start);
00038                 if (neededSize <= Limit - buffer)
00039                 {
00040                     memmove(buffer, Start, End - Start);
00041                     End = buffer + (End - Start);
00042                     Start = buffer;
00043                 }
00044             }
00045             else
00046             {
00047                 char *newBuf;
00048                 int bufferSize = Limit - Start;
00049                 if ( bufferSize == 0 )
00050                 {
00051                     bufferSize = InitBufferSize;
00052                 }
00053                 do
00054                 {
00055                     bufferSize *= 2;
00056                 } while (bufferSize < neededSize);
00057                 //newBuf = malloc(bufferSize);
00058                 try
00059                 {
00060                     Internal.reserve(bufferSize);
00061                     newBuf = &Internal[0];
00062                 }
00063                 catch(...)
00064                 {
00065                     //errorCode = NoMemoryError;
00066                     return 0;
00067                 }
00068                 Limit = newBuf + bufferSize;
00069             }
00070             if (Start)
00071             {
00072                 memcpy(newBuf, Start, End - Start);
00073             }
00074             End = newBuf + (End - Start);
00075             Start = /*buffer ==*/ newBuf;
00076         }
00077     }
00078     gdcm_assert( (int)Internal.capacity() >= len );
00079     return End;
00080 }

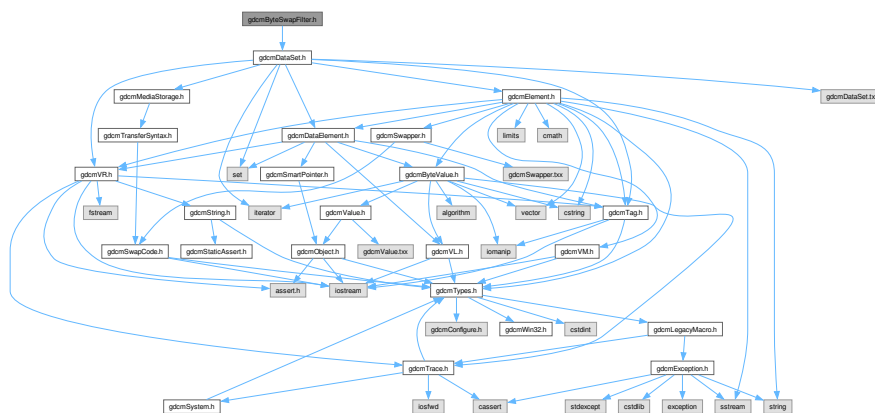
```

```
00088
00089     void UpdatePosition() {}
00090     void ShiftEnd(int len) {
00091         End += len;
00092     }
00093     const char *GetStart() const {
00094         return Start;
00095     }
00096
00097 private:
00098     typedef std::vector<char> CharVector;
00099     const char *Start;
00100     char *End;
00101     const char *Limit;
00102     CharVector Internal;
00103 };
00104
00105 } // end namespace gdcn
00106
00107 #endif //GDCMBYTEBUFFER_H
```

13.115 gdcMByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
```

Include dependency graph for `gdcMByteSwapFilter.h`:



Classes

- class `gdcm::ByteSwapFilter`
`ByteSwapFilter`.

Namespaces

- namespace `gdcm`

13.116 gdcmByteSwapFilter.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMBYTESWAPFILTER_H
00015 #define GDCMBYTESWAPFILTER_H
00016
00017 #include "gdcmDataSet.h"
00018
00019 namespace gdcm
00020 {
00021
00022     class GDCM_EXPORT ByteSwapFilter
00023     {
00024     public:
00025         ByteSwapFilter(DataSet& ds):DS(ds),ByteSwapTag(false) {}
00026         ~ByteSwapFilter() = default;
00027         ByteSwapFilter(const ByteSwapFilter &) = delete;
00028         ByteSwapFilter& operator=(const ByteSwapFilter &) = delete;
00029
00030         bool ByteSwap();
00031         void SetByteSwapTag(bool b) { ByteSwapTag = b; }
00032
00033     private:
00034         DataSet &DS;
00035         bool ByteSwapTag;
00036     };
00037
00038 } // end namespace gdcm
00039
00040 #endif //GDCMBYTESWAPFILTER_H

```

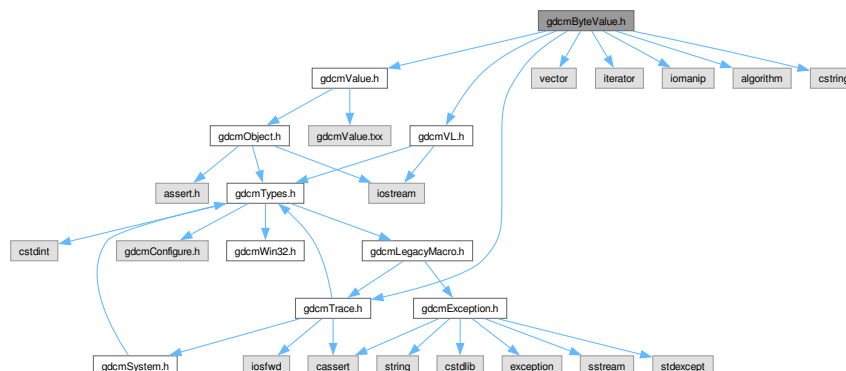
13.117 gdcmByteValue.h File Reference

```

#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
#include <algorithm>
#include <cstring>

```

Include dependency graph for gdcmByteValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ByteValue](#)
Class to represent binary value (array of bytes).

Namespaces

- namespace [gdcm](#)

13.118 gdcmByteValue.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBYTEVALUE_H

```

```

00015 #define GDCMBYTEVALUE_H
00016
00017 #include "gdcmValue.h"
00018 #include "gdcmTrace.h"
00019 #include "gdcmVL.h"
00020
00021 #include <vector>
00022 #include <iterator>
00023 #include <iomanip>
00024 #include <algorithm>
00025 #include <cstring>
00026
00027 namespace gdcm_ns
00028 {
00029 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00030 using namespace gdcm;
00031 #endif
00032 class GDCM_EXPORT ByteValue : public Value
00033 {
00034 public:
00035     ByteValue(const char* array = nullptr, VL const &vl = 0): Length(vl) {
00036         VL bytes_count_to_copy = Length;
00037         if( vl.IsOdd() )
00038         {
00039             gdcmDebugMacro( "Odd length" );
00040             Internal.resize(vl+1);
00041             ++Length;
00042         }
00043         Internal.resize(Length);
00044         if( array )
00045             std::memcpy(Internal.data(), array, bytes_count_to_copy);
00046     }
00047
00048     ByteValue(std::vector<char> &v):Internal(v),Length((uint32_t)v.size()) {}
00049     //ByteValue(std::ostream const &os) {
00050     //    (void)os;
00051     //    gdcm_assert(0); // TODO
00052     //}
00053     ~ByteValue() override {
00054         Internal.clear();
00055     }
00056
00057     // When 'dumping' dicom file we still have some information from
00058     // Either the VR: eg LO (private tag)
00059     void PrintASCII(std::ostream &os, VL maxlength) const;
00060
00061     void PrintHex(std::ostream &os, VL maxlength) const;
00062
00063     // Either from Element Number (== 0x0000)
00064     void PrintGroupLength(std::ostream &os) {
00065         gdcm_assert( Length == 2 );
00066         (void)os;
00067     }
00068
00069     bool IsEmpty() const {
00070         #if 0
00071             if( Internal.empty() ) gdcm_assert( Length == 0 );
00072             return Internal.empty();
00073         #else
00074             return Length == 0;
00075         #endif
00076     }
00077     VL GetLength() const override { return Length; }
00078
00079     VL ComputeLength() const { return Length + Length % 2; }
00080     // Does a reallocation
00081     void SetLength(VL vl) override;
00082
00083     operator const std::vector<char>& () const { return Internal; }
00084
00085     ByteValue &operator=(const ByteValue &val) {
00086         Internal = val.Internal;
00087         Length = val.Length;
00088         return *this;
00089     }
00090
00091     bool operator==(const ByteValue &val) const {
00092         if( Length != val.Length )
00093             return false;
00094         if( Internal == val.Internal )
00095             return true;
00096     }

```

```

00100     return false;
00101   }
00102   bool operator==(const Value &val) const override
00103   {
00104     const ByteValue &bv = dynamic_cast<const ByteValue&>(val);
00105     return Length == bv.Length && Internal == bv.Internal;
00106   }
00107
00108   void Append(ByteValue const & bv);
00109
00110   void Clear() override {
00111     Internal.clear();
00112   }
00113   // Use that only if you understand what you are doing
00114   const char *GetPointer() const {
00115     if(!Internal.empty()) return &Internal[0];
00116     return nullptr;
00117   }
00118   // Use that only if you really understand what you are doing
00119   const void *GetVoidPointer() const {
00120     if(!Internal.empty()) return &Internal[0];
00121     return nullptr;
00122   }
00123   void *GetVoidPointer() {
00124     if(!Internal.empty()) return &Internal[0];
00125     return nullptr;
00126   }
00127   void Fill(char c) {
00128     //if( Internal.empty() ) return;
00129     std::vector<char>::iterator it = Internal.begin();
00130     for(; it != Internal.end(); ++it) *it = c;
00131   }
00132   bool GetBuffer(char *buffer, unsigned long length) const;
00133   bool WriteBuffer(std::ostream &os) const {
00134     if( Length ) {
00135       //gdcm_assert( Internal.size() <= Length );
00136       gdcm_assert( !(Internal.size() % 2) );
00137       os.write(&Internal[0], Internal.size() );
00138     }
00139     return true;
00140   }
00141
00142   template <typename TSwap, typename TType>
00143   std::istream &Read(std::istream &is, bool readvalues = true) {
00144     // If Length is odd we have detected that in SetLength
00145     // and calling std::vector::resize make sure to allocate *AND*
00146     // initialize values to 0 so we are sure to have a \0 at the end
00147     // even in this case
00148     if(Length)
00149     {
00150       if( readvalues )
00151       {
00152         is.read(&Internal[0], Length);
00153         gdcm_assert( Internal.size() == Length || Internal.size() == Length + 1 );
00154         TSwap::SwapArray((TType*)GetVoidPointer(), Internal.size() / sizeof(TType) );
00155       }
00156       else
00157       {
00158         is.seekg(Length, std::ios::cur);
00159       }
00160     }
00161     return is;
00162   }
00163
00164   template <typename TSwap>
00165   std::istream &Read(std::istream &is) {
00166     return Read<TSwap, uint8_t>(is);
00167   }
00168
00169   template <typename TSwap, typename TType>
00170   std::ostream const &Write(std::ostream &os) const {
00171     gdcm_assert( !(Internal.size() % 2) );
00172     if( !Internal.empty() ) {
00173       //os.write(&Internal[0], Internal.size());
00174       std::vector<char> copy = Internal;
00175       TSwap::SwapArray((TType*)(void*)&copy[0], Internal.size() / sizeof(TType) );
00176       os.write(&copy[0], copy.size());
00177     }
00178     return os;
00179   }
00180 }

```

```

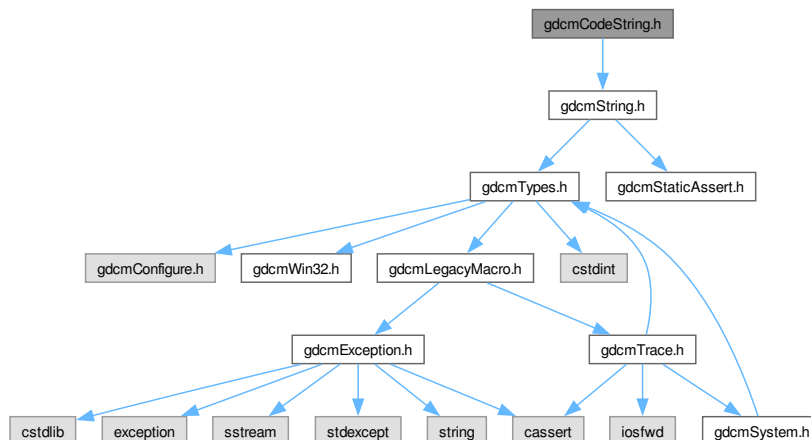
00181
00182 template <typename TSwap>
00183 std::ostream const &Write(std::ostream &os) const {
00184     return Write<TSwap,uint8_t>(os);
00185 }
00186
00193 bool IsPrintable(VL length) const {
00194     gdcM_assert( length <= Length );
00195     for(unsigned int i=0; i<length; i++)
00196     {
00197         if ( i == (length-1) && Internal[i] == '\0') continue;
00198         if ( !( isprint((unsigned char)Internal[i]) || isspace((unsigned char)Internal[i]) ) )
00199         {
00200             //gdcMWarningMacro( "Cannot print :" « i );
00201             return false;
00202         }
00203     }
00204     return true;
00205 }
00206
00208 void PrintPNXML(std::ostream &os) const;
00209 void PrintASCIIXML(std::ostream &os) const;
00210 void PrintHexXML(std::ostream &os) const;
00211 protected:
00212 void Print(std::ostream &os) const override {
00213     // This is perfectly valid to have a Length = 0 , so we cannot check
00214     // the length for printing
00215     if( !Internal.empty() )
00216     {
00217         if( IsPrintable(Length) )
00218         {
00219             // WARNING: Internal.end() != Internal.begin()+Length
00220             std::vector<char>::size_type length = Length;
00221             if( Internal.back() == 0 ) --length;
00222             std::copy(Internal.begin(), Internal.begin()+length,
00223                 std::ostream_iterator<char>(os));
00224         }
00225         else
00226             os « "Loaded:" « Internal.size();
00227     }
00228     else
00229     {
00230         //os « "Not Loaded";
00231         os « "(no value available)";
00232     }
00233 }
00234 /*
00235 //Introduce check for invalid XML characters
00236 friend std::ostream& operator«(std::ostream &os,const char c);
00237 */
00238
00239 void SetLengthOnly(VL vl) override {
00240     Length = vl;
00241 }
00242
00243 private:
00244     std::vector<char> Internal;
00245
00246     // WARNING Length IS NOT Internal.size() some *featured* DICOM
00247     // implementation define odd length, we always load them as even number
00248     // of byte, so we need to keep the right Length
00249     VL Length;
00250 };
00251
00252 } // end namespace gdcM_ns
00253
00254 #endif //GDCMBYTEVALUE_H

```


13.119 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



Classes

- class [gdcm::CodeString](#)
[CodeString](#).

Namespaces

- namespace [gdcm](#)

Functions

- bool [gdcm::operator!=](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- bool [gdcm::operator==](#) (const [CodeString](#) &ref, const [CodeString](#) &cs)

13.120 gdcmCodeString.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.

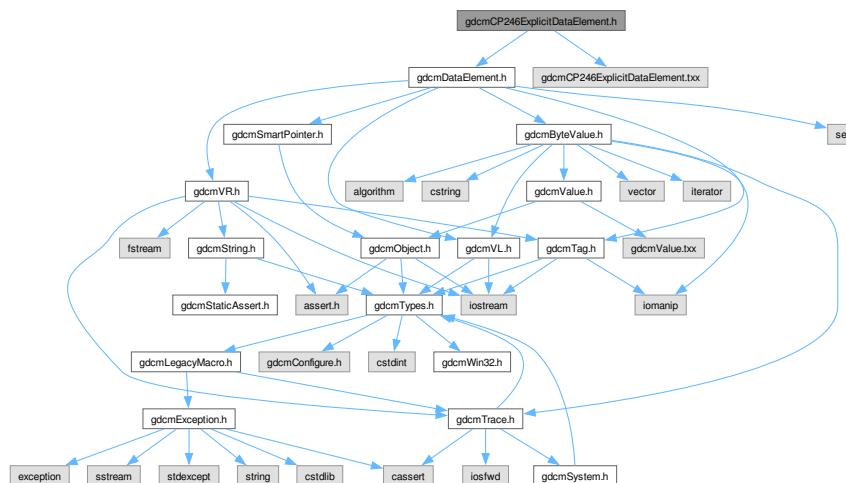
```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMCODESTRING_H
00015 #define GDCMCODESTRING_H
00016
00017 #include "gdcmString.h"
00018
00019 namespace gdcm
00020 {
00021
00022 // Note to myself: because not all wrapped language support exception
00023 // we could not support throwing an exception during object construction.
00024 class GDCM_EXPORT CodeString
00025 {
00026 friend std::ostream& operator<< (std::ostream& os, const CodeString& str);
00027 friend bool operator==(const CodeString &ref, const CodeString& cs);
00028 friend bool operator!=(const CodeString &ref, const CodeString& cs);
00029 typedef String<'\',16> InternalClass;
00030 public:
00031 typedef InternalClass::value_type value_type;
00032 typedef InternalClass::pointer pointer;
00033 typedef InternalClass::reference reference;
00034 typedef InternalClass::const_reference const_reference;
00035 typedef InternalClass::size_type size_type;
00036 typedef InternalClass::difference_type difference_type;
00037 typedef InternalClass::iterator iterator;
00038 typedef InternalClass::const_iterator const_iterator;
00039 typedef InternalClass::reverse_iterator reverse_iterator;
00040 typedef InternalClass::const_reverse_iterator const_reverse_iterator;
00041
00042 CodeString(): Internal() {}
00043 CodeString(const value_type* s): Internal(s) { Internal = Internal.Trim(); }
00044 CodeString(const value_type* s, size_type n): Internal(s, n) {
00045 Internal = Internal.Trim(); }
00046 CodeString(const InternalClass& s, size_type pos=0, size_type n=InternalClass::npos):
00047 Internal(s, pos, n) { Internal = Internal.Trim(); }
00048
00049 bool IsValid() const;
00050
00051 std::string GetAsString() const {
00052 return Internal;
00053 }
00054
00055 size_type Size() const { return Internal.size(); }
00056
00057 protected:
00058 std::string TrimInternal() const {
00059 return Internal.Trim();
00060 }
00061
00062 private:
00063 String<'\',16> Internal;
00064 };
00065
00066 inline std::ostream& operator<< (std::ostream& os, const CodeString& str)
00067 {
00068 os << str.Internal;
00069 return os;
00070 }
00071
00072 inline bool operator==(const CodeString &ref, const CodeString& cs)
00073 {
00074 return ref.Internal == cs.Internal;
00075 }
00076
00077 inline bool operator!=(const CodeString &ref, const CodeString& cs)
00078 {
00079 return ref.Internal != cs.Internal;
00080 }
00081
00082 } // end namespace gdcm
00083
00084 #endif //GDCMCODESTRING_H

```

```
#include "gdcmDataElement.h"
#include "gdcmCP246ExplicitDataElement.txx"
Include dependency graph for gdcmCP246ExplicitDataElement.h:
```



- class `gdc::CP246ExplicitDataElement`
Class to read/write a `DataElement` as CP246Explicit Data `Element`.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*
00014  #ifndef GDCMCP246EXPLICITDATAELEMENT_H

```

```

00015 #define GDCMCP246EXPLICITDATAELEMENT_H
00016
00017 #include "gdcmDataElement.h"
00018
00019 namespace gdcm
00020 {
00021 // Data Element (CP246Explicit)
00022 class GDCM_EXPORT CP246ExplicitDataElement : public DataElement
00023 {
00024 public:
00025 VL GetLength() const;
00026
00027 template <typename TSwap>
00028 std::istream &Read(std::istream &is);
00029
00030 template <typename TSwap>
00031 std::istream &ReadPreValue(std::istream &is);
00032
00033 template <typename TSwap>
00034 std::istream &ReadValue(std::istream &is, bool readvalues = true);
00035
00036 template <typename TSwap>
00037 std::istream &ReadWithLength(std::istream &is, VL & length);
00038
00039 // PURPOSELY do not provide an implementation for writing !
00040 //template <typename TSwap>
00041 //const std::ostream &Write(std::ostream &os) const;
00042 };
00043
00044 } // end namespace gdcm
00045
00046 #include "gdcmCP246ExplicitDataElement.txx"
00047
00048 #endif //GDCMCP246EXPLICITDATAELEMENT_H

```

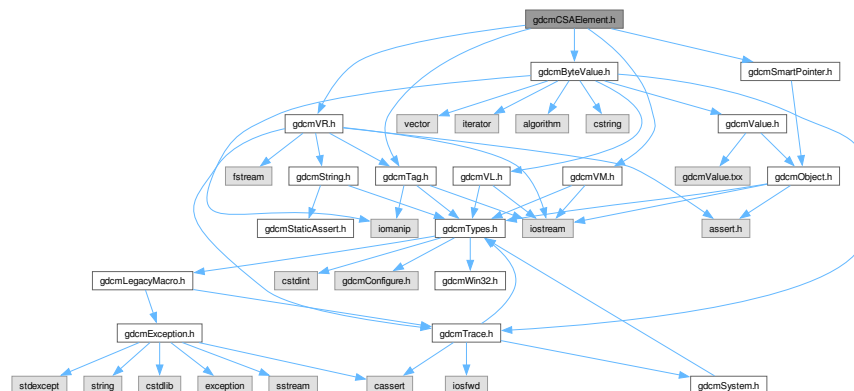
13.123 gdcmCSAElement.h File Reference

```

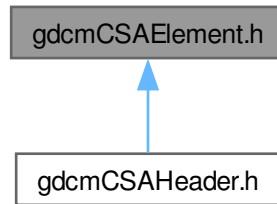
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for gdcmCSAElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::CSAElement](#)
Class to represent a CSA [Element](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)`

13.124 gdcmCSAElement.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCSAELEMENT_H
00015 #define GDCMCSAELEMENT_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmVM.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmByteValue.h"
00021 #include "gdcmSmartPointer.h"

```

```

00022
00023 namespace gdcmm
00024 {
00025 class GDCM_EXPORT CSAElement
00026 {
00027 public:
00028   CSAElement(unsigned int kf = 0):KeyField(kf),SyngoDTField(0),NoOfItemsField(0) {}
00029
00030   friend std::ostream& operator<<(std::ostream &os, const CSAElement &val);
00031
00032   unsigned int GetKey() const { return KeyField; }
00033   void SetKey(unsigned int key) { KeyField = key; }
00034
00035   const char *GetName() const { return NameField.c_str(); }
00036   void SetName(const char *name) { NameField = name; }
00037
00038   const VM& GetVM() const { return ValueMultiplicityField; }
00039   void SetVM(const VM &vm) { ValueMultiplicityField = vm; }
00040
00041   VR const &GetVR() const { return VRField; }
00042   void SetVR(VR const &vr) { VRField = vr; }
00043
00044   unsigned int GetSyngoDT() const { return SyngoDTField; }
00045   void SetSyngoDT(unsigned int syngodt) { SyngoDTField = syngodt; }
00046
00047   unsigned int GetNoOfItems() const { return NoOfItemsField; }
00048   void SetNoOfItems(unsigned int items) { NoOfItemsField = items; }
00049
00050   Value const &GetValue() const { return *DataField; }
00051   Value &GetValue() { return *DataField; }
00052   void SetValue(Value const &vl) {
00053     //gdcmm_assert( DataField == 0 );
00054     DataField = vl;
00055   }
00056   bool IsEmpty() const { return DataField == nullptr; }
00057
00058   void SetByteValue(const char *array, VL length)
00059   {
00060     ByteValue *bv = new ByteValue(array,length);
00061     SetValue( *bv );
00062   }
00063   const ByteValue* GetByteValue() const {
00064     // Get the raw pointer from the gdcmm::SmartPointer
00065     const ByteValue *bv = dynamic_cast<const ByteValue*>(DataField.GetPointer());
00066     return bv; // Will return NULL if not ByteValue
00067   }
00068
00069   CSAElement(const CSAElement &_val)
00070   {
00071     if( this != &_val)
00072     {
00073       *this = _val;
00074     }
00075   }
00076
00077   bool operator<(const CSAElement &de) const
00078   {
00079     return GetKey() < de.GetKey();
00080   }
00081   CSAElement &operator=(const CSAElement &de)
00082   = default;
00083
00084   bool operator==(const CSAElement &de) const
00085   {
00086     return KeyField == de.KeyField
00087        && NameField == de.NameField
00088        && ValueMultiplicityField == de.ValueMultiplicityField
00089        && VRField == de.VRField
00090        && SyngoDTField == de.SyngoDTField
00091        //&& ValueField == de.ValueField;
00092        ;
00093   }
00094
00095 protected:
00096   unsigned int KeyField;
00097   std::string NameField;
00098   VM ValueMultiplicityField;
00099   VR VRField;
00100   unsigned int SyngoDTField;
00101   unsigned int NoOfItemsField;
00102   typedef SmartPointer<Value> DataPtr;

```

```

00118   DataPtr DataField;
00119 };
00120 //-----
00121 inline std::ostream& operator<<(std::ostream &os, const CSAElement &val)
00122 {
00123     os << val.KeyField;
00124     os << " - " << val.NameField;
00125     os << " VM " << val.ValueMultiplicityField;
00126     os << " VR " << val.VRField;
00127     os << " SyngoDT " << val.SyngoDTField;
00128     os << " NoOfItems " << val.NoOfItemsField;
00129     os << " Data ";
00130     if( val.DataField )
00131     {
00132         //val.DataField->Print( os << " " );
00133         const ByteValue * bv = dynamic_cast<ByteValue*>(&*val.DataField);
00134         gdcm_assert( bv );
00135         const char * p = bv->GetPointer();
00136         std::string str(p, p + bv->GetLength() );
00137         if( val.ValueMultiplicityField == VM::VM1 )
00138         {
00139             os << " " << str.c_str() << " ";
00140         }
00141         else
00142         {
00143             std::istringstream is( str );
00144             std::string s;
00145             bool sep = false;
00146             while( std::getline(is, s, '\\') )
00147             {
00148                 if( sep )
00149                 {
00150                     os << '\\';
00151                 }
00152                 sep = true;
00153                 os << " " << s.c_str() << " ";
00154             }
00155             //bv->Print( os << " " );
00156             //os << " ";
00157         }
00158     }
00159     return os;
00160 }
00161
00162 } // end namespace gdcm
00163
00164 #endif //GDCMCSAELEMENT_H

```

13.125 gdcmCSAHeader.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
#include "gdcmMrProtocol.h"

```

- class `gdcm::CSAHeader`
Class for `CSAHeader`.

- namespace `gdcm`

- `std::ostream & gdcmm::operator<< (std::ostream &os, const CSAHeader &d)`

[Go to the documentation of this file.](#)

Generated by Doxygen


```

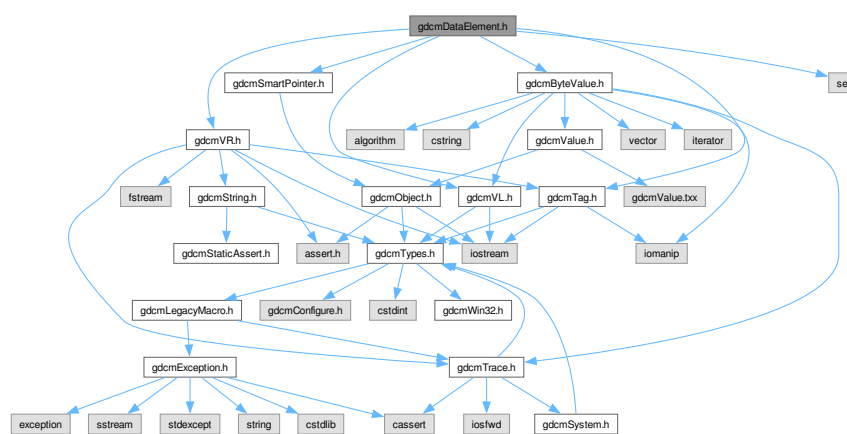
00018 #include "gdcmDataSet.h"
00019 #include "gdcmCSAElement.h"
00020 #include "gdcmMrProtocol.h"
00021
00022 namespace gdcm
00023 {
00024 /*
00025  * Everything done in this code is for the sole purpose of writing interoperable
00026  * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00027  * If you believe anything in this code violates any law or any of your rights,
00028  * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00029  * find a solution.
00030  */
00031 //-----
00032
00033 class DataElement;
00034 class PrivateTag;
00063 class GDCM_EXPORT CSAHeader
00064 {
00065 friend std::ostream& operator«(std::ostream &_os, const CSAHeader &d);
00066 public :
00067 CSAHeader():InternalDataSet(),InternalType(UNKNOWN),InterfileData(nullptr) {}
00068 ~CSAHeader() = default;
00069
00071 typedef enum {
00072     UNKNOWN = 0,
00073     SV10,
00074     NOMAGIC,
00075     DATASET_FORMAT,
00076     INTERFILE,
00077     ZEROED_OUT
00078 } CSAHeaderType;
00079
00081 bool LoadFromDataElement(DataElement const &de);
00082
00084 void Print(std::ostream &os) const;
00085
00087 const DataSet& GetDataSet() const { return InternalDataSet; }
00088
00090 const char * GetInterfile() const { return InterfileData; }
00091
00094 CSAHeaderType GetFormat() const;
00095
00098 static const PrivateTag & GetCSAImageHeaderInfoTag();
00099
00102 static const PrivateTag & GetCSASeriesHeaderInfoTag();
00103
00106 static const PrivateTag & GetCSADataInfo();
00107
00110 const CSAElement &GetCSAElementByName(const char *name);
00111
00114 bool FindCSAElementByName(const char *name);
00115
00117 bool GetMrProtocol( const DataSet & ds, MrProtocol & mrProtocol );
00118
00119 protected:
00120 const CSAElement& GetCSAEEnd() const;
00121
00122 private:
00123 std::set<CSAElement> InternalCSADataset;
00124 DataSet InternalDataSet;
00125 CSAHeaderType InternalType;
00126 Tag DataElementTag;
00127 static CSAElement CSAEEnd;
00128 const char *InterfileData;
00129 };
00130 //-----
00131 inline std::ostream& operator«(std::ostream &os, const CSAHeader &d)
00132 {
00133     d.Print( os );
00134     return os;
00135 }
00136
00137 } // end namespace gdcm
00138 //-----
00139 #endif //GDCMCSAHEADER_H

```

13.127 gdcmDataElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include <set>
```

Include dependency graph for gdcmDataElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::DataElement](#)
Class to represent a Data [Element](#) either Implicit or Explicit.

Namespaces

- namespace [gdcm](#)

Functions

- bool [gdcm::operator!=](#) (const [DataElement](#) &lhs, const [DataElement](#) &rhs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [DataElement](#) &val)

13.128 gdcmDataElement.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDATAELEMENT_H
00015 #define GDCMDATAELEMENT_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmVL.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmByteValue.h"
00021 #include "gdcmSmartPointer.h"
00022
00023 #include <set>
00024
00025 namespace gdcm_ns
00026 {
00027 // Data Element
00028 // Contains multiple fields:
00029 // -> Tag
00030 // -> Optional VR (Explicit Transfer Syntax)
00031 // -> ValueLength
00032 // -> Value
00033 // TODO: This class SHOULD be pure virtual. I don't want a user
00034 // to shoot himself in the foot.
00035
00036 class SequenceOfItems;
00037 class SequenceOfFragments;
00038 class GDCM_EXPORT DataElement
00039 {
00040 public:
00041 DataElement(const Tag& t = Tag(0), const VL& vl = 0, const VR &vr =
00042 VR::INVALID):TagField(t),ValueLengthField(vl),VRField(vr),ValueField(nullptr) {}
00043 //DataElement( Attribute const &att );
00044
00045 friend std::ostream& operator<<(std::ostream &_os, const DataElement &_val);
00046
00047 const Tag& GetTag() const { return TagField; }
00048 Tag& GetTag() { return TagField; }
00049 void SetTag(const Tag &t) { TagField = t; }
00050
00051 const VL& GetVL() const { return ValueLengthField; }
00052 VL& GetVL() { return ValueLengthField; }
00053 void SetVL(const VL &vl) { ValueLengthField = vl; }
00054 void SetVLToUndefined();
00055
00056 VR const &GetVR() const { return VRField; }
00057 void SetVR(VR const &vr) {
00058 if( vr.IsVRFile() )
00059 VRField = vr;
00060 }
00061
00062 Value const &GetValue() const { gdcmAssertAlwaysMacro(ValueField); return *ValueField; }
00063 Value &GetValue() {
00064 gdcmAssertAlwaysMacro(ValueField);
00065 return *ValueField;
00066 }
00067 void SetValue(Value const & vl) {
00068 //gdcm_assert( ValueField == 0 );
00069 ValueField = vl;
00070 ValueLengthField = vl.GetLength();
00071 }
00072 bool IsEmpty() const { return ValueField == nullptr || (GetByteValue() && GetByteValue()->IsEmpty()); }
00073

```

```

00109 void Empty() { ValueField = nullptr; ValueLengthField = 0; }
00110
00111 void Clear()
00112 {
00113     TagField = 0;
00114     VRField = VR::INVALID;
00115     ValueField = nullptr;
00116     ValueLengthField = 0;
00117 }
00118
00119 // Helper:
00120 void SetByteValue(const char *array, VL length)
00121 {
00122     ByteValue *bv = new ByteValue(array,length);
00123     SetValue( *bv );
00124 }
00125
00126 const ByteValue* GetByteValue() const {
00127     // Get the raw pointer from the gdcm::SmartPointer
00128     const ByteValue *bv = dynamic_cast<const ByteValue*>(ValueField.GetPointer());
00129     return bv; // Will return NULL if not ByteValue
00130 }
00131
00132 SmartPointer<SequenceOfItems> GetValueAsSQ() const;
00133
00134 const SequenceOfFragments* GetSequenceOfFragments() const;
00135 SequenceOfFragments* GetSequenceOfFragments();
00136
00137 bool IsUndefinedLength() const {
00138     return ValueLengthField.IsUndefined();
00139 }
00140
00141 DataElement(const DataElement &_val)
00142 {
00143     if( this != &_amp;_val)
00144     {
00145         *this = _val;
00146     }
00147 }
00148
00149 bool operator<(const DataElement &de) const
00150 {
00151     return GetTag() < de.GetTag();
00152 }
00153
00154 DataElement &operator=(const DataElement &)
00155 = default;
00156
00157 bool operator==(const DataElement &de) const
00158 {
00159     bool b = TagField == de.TagField
00160         && ValueLengthField == de.ValueLengthField
00161         && VRField == de.VRField;
00162     if( !ValueField && !de.ValueField )
00163     {
00164         return b;
00165     }
00166     if( ValueField && de.ValueField )
00167     {
00168         return b && (*ValueField == *de.ValueField);
00169     }
00170     // ValueField != de.ValueField
00171     return false;
00172 }
00173
00174 // The following functionalities are dependent on:
00175 // # The Transfer Syntax: Explicit or Implicit
00176 // # The Byte encoding: Little Endian / Big Endian
00177
00178 /*
00179  * The following was inspired by a C++ idiom: Curiously Recurring Template Pattern
00180  * Ref: http://en.wikipedia.org/wiki/Curiously\_Recurring\_Template\_Pattern
00181  * The typename TDE is typically a derived class *without* any data
00182  * while TSwap is a simple template parameter to achieve byteswapping (and allow factorization of
00183  * highly identical code)
00184  */
00185 template <typename TDE>
00186 VL GetLength() const {
00187     return static_cast<const TDE*>(this)->GetLength();
00188 }
00189
00190 template <typename TDE, typename TSwap>
00191 std::istream &Read(std::istream &is) {

```

```

00207     return static_cast<TDE*>(this)->template Read<TSwap>(is);
00208 }
00209
00210 template <typename TDE, typename TSwap>
00211 std::istream &ReadOrSkip(std::istream &is, std::set<Tag> const &skiptags) {
00212     (void)skiptags;
00213     return static_cast<TDE*>(this)->template Read<TSwap>(is);
00214 }
00215
00216 template <typename TDE, typename TSwap>
00217 std::istream &ReadPreValue(std::istream &is, std::set<Tag> const &skiptags) {
00218     (void)skiptags;
00219     return static_cast<TDE*>(this)->template ReadPreValue<TSwap>(is);
00220 }
00221 template <typename TDE, typename TSwap>
00222 std::istream &ReadValue(std::istream &is, std::set<Tag> const &skiptags) {
00223     (void)skiptags;
00224     return static_cast<TDE*>(this)->template ReadValue<TSwap>(is);
00225 }
00226 template <typename TDE, typename TSwap>
00227 std::istream &ReadValueWithLength(std::istream &is, VL & length, std::set<Tag> const &skiptags) {
00228     (void)skiptags;
00229     return static_cast<TDE*>(this)->template ReadValueWithLength<TSwap>(is, length);
00230 }
00231
00232 template <typename TDE, typename TSwap>
00233 std::istream &ReadWithLength(std::istream &is, VL &length) {
00234     return static_cast<TDE*>(this)->template ReadWithLength<TSwap>(is,length);
00235 }
00236
00237 template <typename TDE, typename TSwap>
00238 const std::ostream &Write(std::ostream &os) const {
00239     return static_cast<const TDE*>(this)->template Write<TSwap>(os);
00240 }
00241
00242 protected:
00243     Tag TagField;
00244     // This is the value read from the file, might be different from the length of Value Field
00245     VL ValueLengthField; // Can be 0xFFFFFFFF
00246
00247     // Value Representation
00248     VR VRField;
00249     typedef SmartPointer<Value> ValuePtr;
00250     ValuePtr ValueField;
00251
00252     void SetValueFieldLength( VL vl, bool readvalues );
00253 };
00254 //-----
00255 inline std::ostream& operator<<(std::ostream &os, const DataElement &val)
00256 {
00257     os << val.TagField;
00258     os << "\t" << val.VRField;
00259     os << "\t" << val.ValueLengthField;
00260     if( val.ValueField )
00261     {
00262         val.ValueField->Print( os << "\t" );
00263     }
00264     return os;
00265 }
00266
00267 inline bool operator!=(const DataElement& lhs, const DataElement& rhs)
00268 {
00269     return ! ( lhs == rhs );
00270 }
00271
00272 } // end namespace gdcm_ns
00273
00274 #endif //GDCMDATAELEMENT_H

```

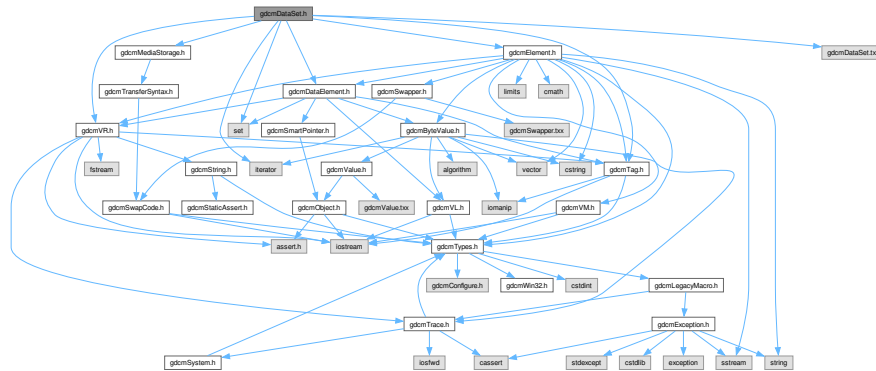
13.129 gdcmDataSet.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmTag.h"
#include "gdcmVR.h"

```

```
#include "gdcmElement.h"
#include "gdcmMediaStorage.h"
#include <set>
#include <iterator>
#include "gdcmDataSet.txx"
Include dependency graph for gdcmDataSet.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::DataElementException`
- class `gdcm::DataSet`

Class to represent a Data Set (which contains Data Elements).

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)`

13.130 gdcmDataSet.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDATASET_H
00015 #define GDCMDATASET_H
00016
00017 #include "gdcmDataElement.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmElement.h"
00021 #include "gdcmMediaStorage.h"
00022
00023 #include <set>
00024 #include <iterator>
00025
00026 namespace gdcm_ns
00027 {
00028 class GDCM_EXPORT DataElementException : public std::exception {};
00029
00030 class PrivateTag;
00031 class GDCM_EXPORT DataSet
00032 {
00033     friend class CSAHeader;
00034 public:
00035     typedef std::set<DataElement> DataElementSet;
00036     typedef DataElementSet::const_iterator ConstIterator;
00037     typedef DataElementSet::iterator Iterator;
00038     typedef DataElementSet::size_type SizeType;
00039     //typedef typename DataElementSet::iterator iterator;
00040     ConstIterator Begin() const { return DES.begin(); }
00041     Iterator Begin() { return DES.begin(); }
00042     ConstIterator End() const { return DES.end(); }
00043     Iterator End() { return DES.end(); }
00044     const DataElementSet &GetDES() const { return DES; }
00045     DataElementSet &GetDES() { return DES; }
00046     void Clear() {
00047         DES.clear();
00048         gdcm_assert( DES.empty() );
00049     }
00050
00051     SizeType Size() const {
00052         return DES.size();
00053     }
00054
00055     void Print(std::ostream &os, std::string const &indent = "") const {
00056         // CT_Phillips_JPEG2K_Decompr_Problem.dcm has a SQ of length == 0
00057         //int s = DES.size();
00058         //gdcm_assert( s );
00059         //std::copy(DES.begin(), DES.end(),
00060         // std::ostream_iterator<DataElement>(os, "\n"));
00061         ConstIterator it = DES.begin();
00062         for( ; it != DES.end(); ++it)
00063         {
00064             os « indent « *it « "\n";
00065         }
00066     }
00067
00068     template <typename TDE>
00069     unsigned int ComputeGroupLength(Tag const &tag) const
00070     {
00071         gdcm_assert( tag.GetElement() == 0x0 );
00072         const DataElement r(tag);
00073         ConstIterator it = DES.find(r);
00074     }
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097

```

```

00098     unsigned int res = 0;
00099     for( ++it; it != DES.end()
00100         && it->GetTag().GetGroup() == tag.GetGroup(); ++it)
00101     {
00102         gdcmm_assert( it->GetTag().GetElement() != 0x0 );
00103         gdcmm_assert( it->GetTag().GetGroup() == tag.GetGroup() );
00104         res += it->GetLength<TDE>();
00105     }
00106     return res;
00107 }
00108
00109 template <typename TDE>
00110 VL GetLength() const {
00111     if( DES.empty() ) return 0;
00112     gdcmm_assert( !DES.empty() );
00113     VL ll = 0;
00114     gdcmm_assert( ll == 0 );
00115     ConstIterator it = DES.begin();
00116     for( ; it != DES.end(); ++it)
00117     {
00118         gdcmm_assert( !(it->GetLength<TDE>().IsUndefined()) );
00119         if ( it->GetTag() != Tag(0xfffe,0xe00d) )
00120         {
00121             ll += it->GetLength<TDE>();
00122         }
00123     }
00124     return ll;
00125 }
00126 void Insert(const DataElement& de) {
00127     // FIXME: there is a special case where a dataset can have value < 0x8, see:
00128     // $ gdcmdump --csa gdcmmData/SIEMENS-JPEG-CorruptFrag.dcm
00129     if( de.GetTag().GetGroup() >= 0x0008 || de.GetTag().GetGroup() == 0x4 )
00130     {
00131         // prevent user error:
00132         if( de.GetTag() == Tag(0xfffe,0xe00d)
00133             || de.GetTag() == Tag(0xfffe,0xe0dd)
00134             || de.GetTag() == Tag(0xfffe,0xe000) )
00135         {
00136             }
00137         else
00138         {
00139             InsertDataElement( de );
00140         }
00141     }
00142     else
00143     {
00144         gdcmmErrorMacro( "Cannot add element with group < 0x0008 and != 0x4 in the dataset: " « de.GetTag() );
00145     }
00146 }
00147 void Replace(const DataElement& de) {
00148     ConstIterator it = DES.find(de);
00149     if( it != DES.end() )
00150     {
00151         // detect loop:
00152         gdcmmAssertAlwaysMacro( &*it != &de );
00153         DES.erase(it);
00154     }
00155     DES.insert(de);
00156 }
00157 void ReplaceEmpty(const DataElement& de) {
00158     ConstIterator it = DES.find(de);
00159     if( it != DES.end() && it->IsEmpty() )
00160     {
00161         // detect loop:
00162         gdcmmAssertAlwaysMacro( &*it != &de );
00163         DES.erase(it);
00164     }
00165     DES.insert(de);
00166 }
00167 SizeType Remove(const Tag& tag) {
00168     DataElementSet::size_type count = DES.erase(tag);
00169     gdcmm_assert( count == 0 || count == 1 );
00170     return count;
00171 }
00172 //DataElement& GetDataElement(const Tag &t) {
00173 //    DataElement r(t);
00174 //    Iterator it = DES.find(r);
00175 //    if( it != DES.end() )
00176 //        return *it;
00177 //    return GetDEEnd();

```



```

00187 // }
00188 const DataElement& GetDataElement(const Tag &t) const {
00189     const DataElement r(t);
00190     ConstIterator it = DES.find(r);
00191     if( it != DES.end() )
00192         return *it;
00193     return GetDEEnd();
00194 }
00195 const DataElement& operator[] (const Tag &t) const { return GetDataElement(t); }
00196 const DataElement& operator() (uint16_t group, uint16_t element) const { return GetDataElement( Tag(group,element) ); }
00197
00200 std::string GetPrivateCreator(const Tag &t) const;
00201
00203 PrivateTag GetPrivateTag(const Tag &t) const;
00204
00206 bool FindDataElement(const PrivateTag &t) const;
00208 const DataElement& GetDataElement(const PrivateTag &t) const;
00209
00210 // DUMB: this only search within the level of the current DataSet
00211 bool FindDataElement(const Tag &t) const {
00212     const auto it = GetDataElement(t);
00213     // Return if tag is found
00214     return it != GetDEEnd();
00215 }
00216
00217 // WARNING:
00218 // This only search at the same level as the DataSet is !
00219 const DataElement& FindNextDataElement(const Tag &t) const {
00220     const DataElement r(t);
00221     ConstIterator it = DES.lower_bound(r);
00222     if( it != DES.end() )
00223         return *it;
00224     return GetDEEnd();
00225 }
00226
00228 bool IsEmpty() const { return DES.empty(); }
00229
00230 DataSet& operator=(DataSet const &)
00231 = default;
00232
00233 template <typename TDE, typename TSwap>
00234 std::istream &ReadNested(std::istream &is);
00235
00236 template <typename TDE, typename TSwap>
00237 std::istream &Read(std::istream &is);
00238
00239 template <typename TDE, typename TSwap>
00240 std::istream &ReadUpToTag(std::istream &is, const Tag &t, std::set<Tag> const &skiptags);
00241
00242 template <typename TDE, typename TSwap>
00243 std::istream &ReadUpToTagWithLength(std::istream &is, const Tag &t, std::set<Tag> const &skiptags, VL &length);
00244
00245 template <typename TDE, typename TSwap>
00246 std::istream &ReadSelectedTags(std::istream &is, const std::set<Tag> &tags, bool readvalues = true);
00247 template <typename TDE, typename TSwap>
00248 std::istream &ReadSelectedTagsWithLength(std::istream &is, const std::set<Tag> &tags, VL &length, bool readvalues =
true);
00249
00250 template <typename TDE, typename TSwap>
00251 std::istream &ReadSelectedPrivateTags(std::istream &is, const std::set<PrivateTag> &tags, bool readvalues = true);
00252 template <typename TDE, typename TSwap>
00253 std::istream &ReadSelectedPrivateTagsWithLength(std::istream &is, const std::set<PrivateTag> &tags, VL &length, bool
readvalues = true);
00254
00255 template <typename TDE, typename TSwap>
00256 std::ostream const &Write(std::ostream &os) const;
00257
00258 template <typename TDE, typename TSwap>
00259 std::istream &ReadWithLength(std::istream &is, VL &length);
00260
00261 MediaStorage GetMediaStorage() const;
00262
00263 protected:
00264 /* GetDEEnd is a Win32 only issue, one cannot use a dllexport
00265  * static member data in an inline function, otherwise symbol
00266  * will get reported as missing in any dll using the inlined function
00267  */
00268 const DataElement& GetDEEnd() const;
00269
00270 // This function is not safe, it does not check for the value of the tag
00271 // so depending whether we are getting called from a dataset or file meta header

```

```

00272 // the condition is different
00273 void InsertDataElement(const DataElement& de) {
00274     //if( de.GetTag() == Tag(0xfffe,0xe00d) ) return;
00275     //if( de.GetTag() == Tag(0xfffe,0xe0dd) ) return;
00276 #ifndef NDEBUG
00277     std::pair<Iterator,bool> pr = DES.insert(de);
00278     if( pr.second == false )
00279     {
00280         gdcmlWarningMacro( "DataElement: " « de « " was already found, skipping duplicate entry.\n"
00281         "Original entry kept is: " « *pr.first );
00282     }
00283 #else
00284     DES.insert(de);
00285 #endif
00286     gdcml_assert( de.IsEmpty() || de.GetVL() == de.GetValue().GetLength() );
00287 }
00288
00289 protected:
00290 // Internal function, that will compute the actual Tag (if found) of
00291 // a requested Private Tag (XXXX,YY,"PRIVATE")
00292 Tag ComputeDataElement(const PrivateTag & t) const;
00293
00294 private:
00295     DataElementSet DES;
00296     static DataElement DEEnd;
00297     friend std::ostream& operator<<(std::ostream &_os, const DataSet &);
00298 };
00299 //-----
00300 inline std::ostream& operator<<(std::ostream &os, const DataSet &val)
00301 {
00302     val.Print(os);
00303     return os;
00304 }
00305
00306 #if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
00307 /*
00308  * HACK: I need this temp class to be able to manipulate a std::set from python,
00309  * swig does not support wrapping of simple class like std::set...
00310  */
00311 class SWIGDataSet
00312 {
00313 public:
00314     SWIGDataSet(DataSet &des):Internal(des),it(des.Begin()) {}
00315     const DataElement& GetCurrent() const { return *it; }
00316     void Start() { it = Internal.Begin(); }
00317     bool IsAtEnd() const { return it == Internal.End(); }
00318     void Next() { ++it; }
00319 private:
00320     DataSet & Internal;
00321     DataSet::ConstIterator it;
00322 };
00323 #endif /* SWIG */
00324
00325
00326 } // end namespace gdcml_ns
00327
00328 #include "gdcmlDataSet.txx"
00329
00330 #endif //GDCMDATASET_H

```

13.131 gdcmlDataSetEvent.h File Reference

```

#include "gdcmlEvent.h"
#include "gdcmlDataSet.h"

```

- class `gdcm::DataSetEvent`
`DataSetEvent`.

- namespace `gdcm`

[Go to the documentation of this file.](#)

Generated by Doxygen

Classes

- class [gdcm::Element< TVR, TVM >](#)
[Element](#) class.
- class [gdcm::Element< TVR, VM::VM1_2 >](#)
- class [gdcm::Element< TVR, VM::VM1_n >](#)
- class [gdcm::Element< TVR, VM::VM2_2n >](#)
- class [gdcm::Element< TVR, VM::VM2_n >](#)
- class [gdcm::Element< TVR, VM::VM3_3n >](#)
- class [gdcm::Element< TVR, VM::VM3_4 >](#)
- class [gdcm::Element< TVR, VM::VM3_n >](#)
- class [gdcm::Element< VR::AS, VM::VM5 >](#)
- class [gdcm::Element< VR::OB, VM::VM1 >](#)
- class [gdcm::Element< VR::OW, VM::VM1 >](#)
- class [gdcm::ElementDisableCombinations< TVR, TVM >](#)

A class which is used to produce compile errors for an invalid combination of template parameters.

- class [gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >](#)
- class [gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >](#)
- class [gdcm::EncodingImplementation< VR::VRASCII >](#)
- class [gdcm::EncodingImplementation< VR::VRBINARY >](#)
- struct [gdcm::ignore_char](#)

Namespaces

- namespace [gdcm](#)

Functions

- static int [gdcm::add1](#) (char *buf, int n)
- [ignore_char](#) const [gdcm::backslash](#) ('\\')
- static void [gdcm::clean](#) (char *mant)
- static int [gdcm::doround](#) (char *buf, unsigned int n)
- std::istream & [gdcm::operator>>](#) (std::istream &in, [ignore_char](#) const &ic)
- static int [gdcm::roundat](#) (char *buf, size_t bufLen, unsigned int i, int iexp)
- template<typename Float>
static void [gdcm::x16printf](#) (char *buf, int size, Float f)

13.134 gdcmElement.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```

00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 #ifndef GDCMELEMENT_H
00015 #define GDCMELEMENT_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmVM.h"
00021 #include "gdcmByteValue.h"
00022 #include "gdcmDataElement.h"
00023 #include "gdcmSwapper.h"
00024
00025 #include <string>
00026 #include <vector>
00027 #include <sstream>
00028 #include <limits>
00029 #include <cmath>
00030 #include <cstring>
00031
00032 namespace gdcm_ns
00033 {
00034
00035 // Forward declaration
00041 template<long long T> class EncodingImplementation;
00042
00043
00051 template <long long TVR, int TVM>
00052 class ElementDisableCombinations {};
00053 template <>
00054 class ElementDisableCombinations<VR::OB, VM::VM1_n> {};
00055 template <>
00056 class ElementDisableCombinations<VR::OW, VM::VM1_n> {};
00057 // Make it impossible to compile these other cases
00058 template <int TVM>
00059 class ElementDisableCombinations<VR::OB, TVM>;
00060 template <int TVM>
00061 class ElementDisableCombinations<VR::OW, TVM>;
00062
00068 template<long long TVR, int TVM>
00069 class Element
00070 {
00071     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, TVM> ) };
00072 public:
00073     typename VRToType<TVR>::Type Internal[VMToLength<TVM>::Length];
00074     typedef typename VRToType<TVR>::Type Type;
00075
00076     static VR GetVR() { return (VR::VRType)TVR; }
00077     static VM GetVM() { return (VM::VMType)TVM; }
00078
00079     unsigned long GetLength() const {
00080         return VMToLength<TVM>::Length;
00081     }
00082     // Implementation of Print is common to all Mode (ASCII/Binary)
00083     // TODO: Can we print a \ when in ASCII...well I don't think so
00084     // it would mean we used a bad VM then, right?
00085     void Print(std::ostream &_os) const {
00086         _os << Internal[0]; // VM is at least guarantee to be one
00087         for(int i=1; i<VMToLength<TVM>::Length; ++i)
00088             _os << ", " << Internal[i];
00089     }
00090
00091     const typename VRToType<TVR>::Type *GetValues() const {
00092         return Internal;
00093     }
00094     const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
00095         gdcm_assert( idx < VMToLength<TVM>::Length );
00096         return Internal[idx];
00097     }
00098     typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
00099         gdcm_assert( idx < VMToLength<TVM>::Length );
00100         return Internal[idx];
00101     }
00102     typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
00103         return GetValue(idx);
00104     }
00105     void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
00106         gdcm_assert( idx < VMToLength<TVM>::Length );
00107         Internal[idx] = v;

```

```

00108 }
00109
00110 void SetFromDataElement(DataElement const &de) {
00111     const ByteValue *bv = de.GetByteValue();
00112     if( !bv ) return;
00113 #ifdef GDCM_WORDS_BIGENDIAN
00114     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00115 #else
00116     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00117 #endif
00118     {
00119         Set(de.GetValue());
00120     }
00121     else
00122     {
00123         SetNoSwap(de.GetValue());
00124     }
00125 }
00126
00127 DataElement GetAsDataElement() const {
00128     DataElement ret;
00129     std::ostream os;
00130     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00131         GetLength(),os);
00132     ret.SetVR( (VR::VRType)TVR );
00133     gdcm_assert( ret.GetVR() != VR::SQ );
00134     if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00135     {
00136         if( GetVR() != VR::UI )
00137         {
00138             if( os.str().size() % 2 )
00139             {
00140                 os << " ";
00141             }
00142         }
00143     }
00144     VL::Type osStrSize = (VL::Type)os.str().size();
00145     ret.SetByteValue( os.str().c_str(), osStrSize );
00146
00147     return ret;
00148 }
00149
00150 void Read(std::istream &_is) {
00151     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00152         GetLength(),_is);
00153 }
00154 void Write(std::ostream &_os) const {
00155     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00156         GetLength(),_os);
00157 }
00158
00159 // FIXME: remove this function
00160 // this is only used in gdcm::SplitMosaicFilter / to pass value of a CSAElement
00161 void Set(Value const &v) {
00162     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00163     if( bv ) {
00164         //memcpy(Internal, bv->GetPointer(), bv->GetLength());
00165         std::stringstream ss;
00166         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00167         ss.str( s );
00168         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00169             GetLength(),ss);
00170     }
00171 }
00172 protected:
00173 void SetNoSwap(Value const &v) {
00174     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00175     gdcm_assert( bv ); // That would be bad...
00176     //memcpy(Internal, bv->GetPointer(), bv->GetLength());
00177     std::stringstream ss;
00178     std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00179     ss.str( s );
00180     EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00181         GetLength(),ss);
00182 }
00183 };
00184
00185 struct ignore_char {
00186     ignore_char(char c): m_char(c) {}
00187     char m_char;
00188 };

```

```

00189 ignore_char const backslash('\\');
00190
00191 inline std::istream& operator» (std::istream& in, ignore_char const& ic) {
00192     if (!in.eof())
00193         in.clear(in.rdstate() & ~std::ios_base::failbit);
00194     if (in.get() != ic.m_char)
00195         in.setstate(std::ios_base::failbit);
00196     return in;
00197 }
00198
00199
00200 // Implementation to perform formatted read and write
00201 template<> class EncodingImplementation<VR::VRASCII> {
00202 public:
00203     template<typename T> // FIXME this should be VRToType<TVR>::Type
00204     static inline void ReadComputeLength(T* data, unsigned int &length,
00205                                         std::istream &_is) {
00206         gdcmm_assert( data );
00207         //gdcmm_assert( length ); // != 0
00208         length = 0;
00209         gdcmm_assert( _is );
00210 #if 0
00211         char sep;
00212         while( _is » data[length++] )
00213         {
00214             // Get the separator in between the values
00215             gdcmm_assert( _is );
00216             _is.get(sep);
00217             gdcmm_assert( sep == '\\' || sep == ' ' ); // FIXME: Bad use of assert
00218             if( sep == ' ' ) length--; // FIXME
00219         }
00220 #else
00221         while( _is » std::ws » data[length++] » std::ws » backslash )
00222         {
00223         }
00224 #endif
00225     }
00226
00227     template<typename T> // FIXME this should be VRToType<TVR>::Type
00228     static inline void Read(T* data, unsigned long length,
00229                             std::istream &_is) {
00230         gdcmm_assert( data );
00231         gdcmm_assert( length ); // != 0
00232         gdcmm_assert( _is );
00233         // FIXME BUG: what if » operation fails ?
00234         // gdcmmData/MR00010001.dcm / SpacingBetweenSlices
00235         _is » std::ws » data[0];
00236         char sep;
00237         //std::cout « "GetLength: " « af->GetLength() « std::endl;
00238         for(unsigned long i=1; i<length;++i) {
00239             //gdcmm_assert( _is );
00240             // Get the separator in between the values
00241             _is » std::ws » sep; // _is.get(sep);
00242             //gdcmm_assert( sep == '\\' ); // FIXME: Bad use of assert
00243             _is » std::ws » data[i];
00244         }
00245     }
00246
00247     template<typename T>
00248     static inline void ReadNoSwap(T* data, unsigned long length,
00249                                   std::istream &_is) {
00250         Read(data,length,_is);
00251     }
00252
00253     template<typename T>
00254     static inline void Write(const T* data, unsigned long length,
00255                              std::ostream &_os) {
00256         gdcmm_assert( data );
00257         gdcmm_assert( length );
00258         gdcmm_assert( _os );
00259         _os « data[0];
00260         for(unsigned long i=1; i<length; ++i) {
00261             gdcmm_assert( _os );
00262             _os « "\\ " « data[i];
00263         }
00264     };
00265
00266 // #define VRDS16ILLEGAL
00267
00268 #ifdef VRDS16ILLEGAL
00269 template < typename Float >

```



```

00270 std::string to_string ( Float data ) {
00271     std::stringstream in;
00272     // in.imbue(std::locale::classic()); // This is not required AFAIK
00273     int const digits =
00274         static_cast< int >(
00275             - std::log( std::numeric_limits<Float>::epsilon() )
00276             / static_cast< Float >( std::log( 10.0 ) ) );
00277     if ( in « std::dec « std::setprecision( /*2+*/digits) « data ) {
00278         return ( in.str() );
00279     } else {
00280         throw "Impossible Conversion"; // should not happen ...
00281     }
00282 }
00283 #else
00284 // http://stackoverflow.com/questions/32631178/writing-ieee-754-1985-double-as-ascii-on-a-limited-16-bytes-string
00285
00286 static inline void clean(char *mant) {
00287     char *ix = mant + strlen(mant) - 1;
00288     while(('0' == *ix) && (ix > mant)) {
00289         *ix-- = '\0';
00290     }
00291     if ( '.' == *ix ) {
00292         *ix = '\0';
00293     }
00294 }
00295
00296 static int add1(char *buf, int n) {
00297     if (n < 0) return 1;
00298     if (buf[n] == '9') {
00299         buf[n] = '0';
00300         return add1(buf, n-1);
00301     }
00302     else {
00303         buf[n] = (char)(buf[n] + 1);
00304     }
00305     return 0;
00306 }
00307
00308 static int doround(char *buf, unsigned int n) {
00309     char c;
00310     if (n >= strlen(buf)) return 0;
00311     c = buf[n];
00312     buf[n] = 0;
00313     if ((c >= '5') && (c <= '9')) return add1(buf, n-1);
00314     return 0;
00315 }
00316
00317 #if defined(_MSC_VER) && (_MSC_VER < 1900)
00318 #define snprintf __snprintf
00319 #endif
00320
00321 static int roundat(char *buf, size_t bufLen, unsigned int i, int iexp) {
00322     if (doround(buf, i) != 0) {
00323         iexp += 1;
00324         switch(iexp) {
00325             case -2:
00326                 strcpy(buf, ".01");
00327                 break;
00328             case -1:
00329                 strcpy(buf, ".1");
00330                 break;
00331             case 0:
00332                 strcpy(buf, "1.");
00333                 break;
00334             case 1:
00335                 strcpy(buf, "10");
00336                 break;
00337             case 2:
00338                 strcpy(buf, "100");
00339                 break;
00340             default:
00341                 sprintf(buf, bufLen, "1e%d", iexp);
00342         }
00343         return 1;
00344     }
00345     return 0;
00346 }
00347
00348 template < typename Float >
00349 static void x16printf(char *buf, int size, Float f) {
00350     char line[40];

```

```

00351 char *mant = line + 1;
00352 int iexp, lexp, i;
00353 char exp[6];
00354
00355 if (f < 0) {
00356     f = -f;
00357     size -= 1;
00358     *buf++ = '-';
00359 }
00360 snprintf(line, sizeof(line), "%1.16e", f);
00361 if (line[0] == '-') {
00362     f = -f;
00363     size -= 1;
00364     *buf++ = '-';
00365     snprintf(line, sizeof(line), "%1.16e", f);
00366 }
00367 *mant = line[0];
00368 i = (int)strcspn(mant, "eE");
00369 mant[i] = '\0';
00370 iexp = (int)strtol(mant + i + 1, nullptr, 10);
00371 lexp = snprintf(exp, sizeof(exp), "%d", iexp);
00372 if ((iexp >= size) || (iexp < -3)) {
00373     i = roundat(mant, sizeof(line) - 1, size - 1 - lexp, iexp);
00374     if (i == 1) {
00375         strcpy(buf, mant);
00376         return;
00377     }
00378     buf[0] = mant[0];
00379     buf[1] = '-';
00380     strncpy(buf + i + 2, mant + 1, size - 2 - lexp);
00381     buf[size - lexp] = 0;
00382     clean(buf);
00383     strcat(buf, exp);
00384 }
00385 else if (iexp >= size - 2) {
00386     roundat(mant, sizeof(line) - 1, iexp + 1, iexp);
00387     strcpy(buf, mant);
00388 }
00389 else if (iexp >= 0) {
00390     i = roundat(mant, sizeof(line) - 1, size - 1, iexp);
00391     if (i == 1) {
00392         strcpy(buf, mant);
00393         return;
00394     }
00395     memcpy(buf, mant, iexp + 1);
00396     buf[iexp + 1] = '-';
00397     strncpy(buf + iexp + 2, mant + iexp + 1, size - iexp - 1);
00398     buf[size] = 0;
00399     clean(buf);
00400 }
00401 else {
00402     int j;
00403     i = roundat(mant, sizeof(line) - 1, size + 1 + iexp, iexp);
00404     if (i == 1) {
00405         strcpy(buf, mant);
00406         return;
00407     }
00408     buf[0] = '-';
00409     for (j=0; j< -1 - iexp; j++) {
00410         buf[j+1] = '0';
00411     }
00412     memcpy(buf - iexp, mant, size + 1 + iexp);
00413     buf[size] = 0;
00414     clean(buf);
00415 }
00416 }
00417 #if defined(_MSC_VER) && (_MSC_VER < 1900)
00418 #undef snprintf
00419 #endif
00420
00421 #endif
00422
00423 template<> inline void EncodingImplementation<VR::VRASCII>::Write(const double* data, unsigned long length,
00424     std::ostream &_os) {
00424     gdcmm_assert( data );
00425     gdcmm_assert( length );
00426     gdcmm_assert( _os );
00427 #ifdef VRDS16ILLEGAL
00428     _os « to_string(data[0]);
00429 #else
00430     char buf[16+1];

```

```

00431     x16printf(buf, 16, data[0]);
00432     _os « buf;
00433 #endif
00434     for(unsigned long i=1; i<length; ++i) {
00435         gdcm_assert( _os );
00436 #ifdef VRDS16ILLEGAL
00437         _os « "\\\" « to_string(data[i]);
00438 #else
00439         x16printf(buf, 16, data[i]);
00440         _os « "\\\" « buf;
00441 #endif
00442     }
00443 }
00444
00445
00446 // Implementation to perform binary read and write
00447 // TODO rewrite operation so that either:
00448 // #1. dummy implementation use a pointer to Internal and do ++p (faster)
00449 // #2. Actually do some meta programming to unroll the loop
00450 // (no notion of order in VM ...)
00451 template<> class EncodingImplementation<VR::VRBINARY> {
00452 public:
00453     template<typename T> // FIXME this should be VRToType<TVR>::Type
00454     static inline void ReadComputeLength(T* data, unsigned int &length,
00455         std::istream &_is) {
00456         const unsigned int type_size = sizeof(T);
00457         gdcm_assert( data ); // Can we read from pointer ?
00458         //gdcm_assert( length );
00459         length /= type_size;
00460         gdcm_assert( _is ); // Is stream valid ?
00461         _is.read( reinterpret_cast<char*>(data+0), type_size);
00462         for(unsigned long i=1; i<length; ++i) {
00463             gdcm_assert( _is );
00464             _is.read( reinterpret_cast<char*>(data+i), type_size );
00465         }
00466     }
00467     template<typename T>
00468     static inline void ReadNoSwap(T* data, unsigned long length,
00469         std::istream &_is) {
00470         const unsigned int type_size = sizeof(T);
00471         gdcm_assert( data ); // Can we read from pointer ?
00472         gdcm_assert( length );
00473         gdcm_assert( _is ); // Is stream valid ?
00474         _is.read( reinterpret_cast<char*>(data+0), type_size);
00475         for(unsigned long i=1; i<length; ++i) {
00476             if( _is )
00477                 _is.read( reinterpret_cast<char*>(data+i), type_size );
00478         }
00479         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
00480         // _is.GetSwapCode(), length);
00481         //SwapperNoOp::SwapArray(data,length);
00482     }
00483     template<typename T>
00484     static inline void Read(T* data, unsigned long length,
00485         std::istream &_is) {
00486         const unsigned int type_size = sizeof(T);
00487         gdcm_assert( data ); // Can we read from pointer ?
00488         gdcm_assert( length );
00489         gdcm_assert( _is ); // Is stream valid ?
00490         _is.read( reinterpret_cast<char*>(data+0), type_size);
00491         for(unsigned long i=1; i<length; ++i) {
00492             if( _is )
00493                 _is.read( reinterpret_cast<char*>(data+i), type_size );
00494         }
00495         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
00496         // _is.GetSwapCode(), length);
00497         SwapperNoOp::SwapArray(data,length);
00498     }
00499     template<typename T>
00500     static inline void Write(const T* data, unsigned long length,
00501         std::ostream &_os) {
00502         const unsigned int type_size = sizeof(T);
00503         gdcm_assert( data ); // Can we write into pointer ?
00504         gdcm_assert( length );
00505         gdcm_assert( _os ); // Is stream valid ?
00506         //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
00507         // _os.GetSwapCode(), length);
00508         T swappedData = SwapperNoOp::Swap(data[0]);
00509         _os.write( reinterpret_cast<const char*>(&swappedData), type_size);
00510         for(unsigned long i=1; i<length; ++i) {
00511             gdcm_assert( _os );

```

```

00512     swappedData = SwapperNoOp::Swap(data[i]);
00513     _os.write( reinterpret_cast<const char*>(&swappedData), type_size );
00514 }
00515 //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
00516 // _os.GetSwapCode(), length);
00517 }
00518 };
00519
00520 // For particular case for ASCII string
00521 // WARNING: This template explicitly instantiates a particular
00522 // EncodingImplementation THEREFORE it is required to be declared after the
00523 // EncodingImplementation is needs (doh!)
00524 #if 0
00525 template<int TVM>
00526 class Element<TVM>
00527 {
00528 public:
00529     Element(const char array[])
00530     {
00531         unsigned int i = 0;
00532         const char sep = '\\';
00533         std::string sarray = array;
00534         std::string::size_type pos1 = 0;
00535         std::string::size_type pos2 = sarray.find(sep, pos1+1);
00536         while(pos2 != std::string::npos)
00537         {
00538             Internal[i++] = sarray.substr(pos1, pos2-pos1);
00539             pos1 = pos2+1;
00540             pos2 = sarray.find(sep, pos1+1);
00541         }
00542         Internal[i] = sarray.substr(pos1, pos2-pos1);
00543         // Shouldn't we do the contrary, since we know how many separators
00544         // (and default behavior is to discard anything after the VM declared
00545         gdcml_assert( GetLength()-1 == i );
00546     }
00547
00548     unsigned long GetLength() const {
00549         return VMToLength<TVM>::Length;
00550     }
00551     // Implementation of Print is common to all Mode (ASCII/Binary)
00552     void Print(std::ostream &_os) const {
00553         _os << Internal[0]; // VM is at least guarantee to be one
00554         for(int i=1; i<VMToLength<TVM>::Length; ++i)
00555             _os << ", " << Internal[i];
00556     }
00557
00558     void Read(std::istream &_is) {
00559         EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(), _is);
00560     }
00561     void Write(std::ostream &_os) const {
00562         EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), _os);
00563     }
00564 private:
00565     typename String Internal[VMToLength<TVM>::Length];
00566 };
00567
00568 template< int TVM>
00569 class Element<VR::PN, TVM> : public StringElement<TVM>
00570 {
00571     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::PN, TVM> ) };
00572 };
00573 #endif
00574
00575 // Implementation for the undefined length (dynamically allocated array)
00576 template<long long TVR>
00577 class Element<TVR, VM::VM1_n>
00578 {
00579     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM1_n> ) };
00580 public:
00581     // This the way to prevent default initialization
00582     explicit Element() { Internal=nullptr; Length=0; Save = false; }
00583     ~Element() {
00584         if( Save ) {
00585             delete[] Internal;
00586         }
00587         Internal = nullptr;
00588     }
00589
00590     static VR GetVR() { return (VR::VRType)TVR; }
00591     static VM GetVM() { return VM::VM1_n; }
00592

```

```

00593 // Length manipulation
00594 // SetLength should really be protected anyway...all operation
00595 // should go through SetArray
00596 unsigned long GetLength() const { return Length; }
00597 typedef typename VRToType<TVR>::Type Type;
00598
00599 void SetLength(unsigned long len) {
00600     const unsigned int size = sizeof(Type);
00601     if( len ) {
00602         if( len > Length ) {
00603             // perform realloc
00604             gdcm_assert( (len / size) * size == len );
00605             Type *internal = new Type[len / size];
00606             gdcm_assert( Save == false );
00607             Save = true; // ???
00608             if( Internal )
00609             {
00610                 memcpy(internal, Internal, len);
00611                 delete[] Internal;
00612             }
00613             Internal = internal;
00614         }
00615     }
00616     Length = len / size;
00617 }
00618
00619 // If save is set to zero user should not delete the pointer
00620 //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
00621 void SetArray(const Type *array, unsigned long len,
00622     bool save = false) {
00623     if( save ) {
00624         SetLength(len); // realloc
00625         memcpy(Internal, array, len/*sizeof(Type)*/);
00626         gdcm_assert( Save == false );
00627     }
00628     else {
00629         // TODO rewrite this stupid code:
00630         gdcm_assert( Length == 0 );
00631         gdcm_assert( Internal == nullptr );
00632         gdcm_assert( Save == false );
00633         Length = len / sizeof(Type);
00634         //gdcm_assert( (len / sizeof(Type)) * sizeof(Type) == len );
00635         // MR00010001.dcm is a tough kid: 0019.105a is supposed to be VR::FL, VM::VM3 but
00636         // length is 14 bytes instead of 12 bytes. Simply consider value is total garbage.
00637         if( (len / sizeof(Type)) * sizeof(Type) != len ) { Internal = nullptr; Length = 0; }
00638         else Internal = const_cast<Type*>(array);
00639     }
00640     Save = save;
00641 }
00642 void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
00643     gdcm_assert( idx < Length );
00644     Internal[idx] = v;
00645 }
00646 const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
00647     gdcm_assert( idx < Length );
00648     return Internal[idx];
00649 }
00650 typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
00651     //gdcm_assert( idx < Length );
00652     return Internal[idx];
00653 }
00654 typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
00655     return GetValue(idx);
00656 }
00657 void Set(Value const &v) {
00658     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00659     gdcm_assert( bv ); // That would be bad...
00660     if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
00661     {
00662         const Type* array = (const Type*)bv->GetVoidPointer();
00663         if( array ) {
00664             gdcm_assert( array ); // That would be bad...
00665             gdcm_assert( Internal == nullptr );
00666             SetArray(array, bv->GetLength() ); }
00667     }
00668     else
00669     {
00670         std::stringstream ss;
00671         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00672         ss.str( s );
00673         EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,

```

```

00674     GetLength(),ss);
00675 }
00676 }
00677 void SetFromDataElement(DataElement const &de) {
00678     const ByteValue *bv = de.GetByteValue();
00679     if( !bv ) return;
00680 #ifdef GDCM_WORDS_BIGENDIAN
00681     if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
00682 #else
00683     if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
00684 #endif
00685     {
00686         Set(de.GetValue());
00687     }
00688     else
00689     {
00690         SetNoSwap(de.GetValue());
00691     }
00692 }
00693
00694
00695 // Need to be placed after definition of EncodingImplementation<VR::VRASCII>
00696 void WriteASCII(std::ostream &os) const {
00697     return EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), os);
00698 }
00699
00700 // Implementation of Print is common to all Mode (ASCII/Binary)
00701 void Print(std::ostream &_os) const {
00702     gdcml_assert( Length );
00703     gdcml_assert( Internal );
00704     _os << Internal[0]; // VM is at least guarantee to be one
00705     const unsigned long length = GetLength() < 25 ? GetLength() : 25;
00706     for(unsigned long i=1; i<length; ++i)
00707         _os << ", " << Internal[i];
00708 }
00709 void Read(std::istream &_is) {
00710     if( !Internal ) return;
00711     EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
00712         GetLength(),_is);
00713 }
00714 //void ReadComputeLength(std::istream &_is) {
00715 //    if( !Internal ) return;
00716 //    EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(Internal,
00717 //        Length,_is);
00718 // }
00719 void Write(std::ostream &_os) const {
00720     EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00721         GetLength(),_os);
00722 }
00723
00724 DataElement GetAsDataElement() const {
00725     DataElement ret;
00726     ret.SetVR( (VR::VRType)TVR );
00727     gdcml_assert( ret.GetVR() != VR::SQ );
00728     if( Internal )
00729     {
00730         std::ostringstream os;
00731         EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
00732             GetLength(),os);
00733         if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
00734         {
00735             if( GetVR() != VR::UI )
00736             {
00737                 if( os.str().size() % 2 )
00738                 {
00739                     os << " ";
00740                 }
00741             }
00742         }
00743         VL::Type osStrSize = (VL::Type)os.str().size();
00744         ret.SetByteValue( os.str().c_str(), osStrSize );
00745     }
00746     return ret;
00747 }
00748
00749 Element(const Element&_val) {
00750     if( this != &_amp;_val ) {
00751         *this = _val;
00752     }
00753 }
00754

```

```

00755 Element &operator=(const Element &_val) {
00756     Length = 0; // SYTTF
00757     Internal = 0;
00758     SetArray(_val.Internal, _val.Length, true);
00759     return *this;
00760 }
00761 protected:
00762 void SetNoSwap(Value const &v) {
00763     const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
00764     gdcm_assert( bv ); // That would be bad...
00765     if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
00766     {
00767         const Type* array = (const Type*)bv->GetPointer();
00768         if( array ) {
00769             gdcm_assert( array ); // That would be bad...
00770             gdcm_assert( Internal == nullptr );
00771             SetArray(array, bv->GetLength() ); }
00772         }
00773     else
00774     {
00775         std::stringstream ss;
00776         std::string s = std::string( bv->GetPointer(), bv->GetLength() );
00777         ss.str( s );
00778         EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
00779             GetLength(),ss);
00780     }
00781 }
00782
00783 private:
00784 typename VRToType<TVR>::Type *Internal;
00785 unsigned long Length; // unsigned int ??
00786 bool Save;
00787 };
00788
00789 //template <int TVM = VM::VM1_n>
00790 //class Element<VR::OB, TVM > : public Element<VR::OB, VM::VM1_n> {};
00791
00792 // Partial specialization for derivatives of 1-n : 2-n, 3-n ...
00793 template<long long TVR>
00794 class Element<TVR, VM::VM1_2> : public Element<TVR, VM::VM1_n>
00795 {
00796 public:
00797     typedef Element<TVR, VM::VM1_n> Parent;
00798     void SetLength(int len) {
00799         if( len != 1 && len != 2 ) return;
00800         Parent::SetLength(len);
00801     }
00802 };
00803 template<long long TVR>
00804 class Element<TVR, VM::VM2_n> : public Element<TVR, VM::VM1_n>
00805 {
00806     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_n> ) };
00807 public:
00808     typedef Element<TVR, VM::VM1_n> Parent;
00809     void SetLength(int len) {
00810         if( len <= 1 ) return;
00811         Parent::SetLength(len);
00812     }
00813 };
00814 template<long long TVR>
00815 class Element<TVR, VM::VM2_2n> : public Element<TVR, VM::VM2_n>
00816 {
00817     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_2n> ) };
00818 public:
00819     typedef Element<TVR, VM::VM2_n> Parent;
00820     void SetLength(int len) {
00821         if( len % 2 ) return;
00822         Parent::SetLength(len);
00823     }
00824 };
00825 template<long long TVR>
00826 class Element<TVR, VM::VM3_n> : public Element<TVR, VM::VM1_n>
00827 {
00828     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_n> ) };
00829 public:
00830     typedef Element<TVR, VM::VM1_n> Parent;
00831     void SetLength(int len) {
00832         if( len <= 2 ) return;
00833         Parent::SetLength(len);
00834     }
00835 };

```

```

00836 template<long long TVR>
00837 class Element<TVR, VM::VM3_3n> : public Element<TVR, VM::VM3_n>
00838 {
00839     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_3n> ) };
00840 public:
00841     typedef Element<TVR, VM::VM3_n> Parent;
00842     void SetLength(int len) {
00843         if( len % 3 ) return;
00844         Parent::SetLength(len);
00845     }
00846 };
00847 template<long long TVR>
00848 class Element<TVR, VM::VM3_4> : public Element<TVR, VM::VM1_n>
00849 {
00850 public:
00851     typedef Element<TVR, VM::VM1_n> Parent;
00852     void SetLength(int len) {
00853         if( len != 3 && len != 4 ) return;
00854         Parent::SetLength(len);
00855     }
00856 };
00857
00858
00859 //template<int T> struct VRToLength;
00860 //template<> struct VRToLength<VR::AS>
00861 //{ enum { Length = VM::VM1 } };
00862 //template<>
00863 //class Element<VR::AS> : public Element<VR::AS, VRToLength<VR::AS>::Length >
00864
00865 // only 0010 1010 AS 1 Patient's Age
00866 template<>
00867 class Element<VR::AS, VM::VM5>
00868 {
00869     enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::AS, VM::VM5> ) };
00870 public:
00871     char Internal[VMToLength<VM::VM5>::Length * sizeof( VRToType<VR::AS>::Type )];
00872     void Print(std::ostream &_os) const {
00873         _os << Internal;
00874     }
00875     unsigned long GetLength() const {
00876         return VMToLength<VM::VM5>::Length;
00877     }
00878 };
00879
00880
00881 template<>
00882 class Element<VR::OB, VM::VM1> : public Element<VR::OB, VM::VM1_n> {};
00883
00884 // Same for OW:
00885 template<>
00886 class Element<VR::OW, VM::VM1> : public Element<VR::OW, VM::VM1_n> {};
00887
00888
00889 } // namespace gdcn_ns
00890
00891 #endif //GDCMELEMENT_H

```

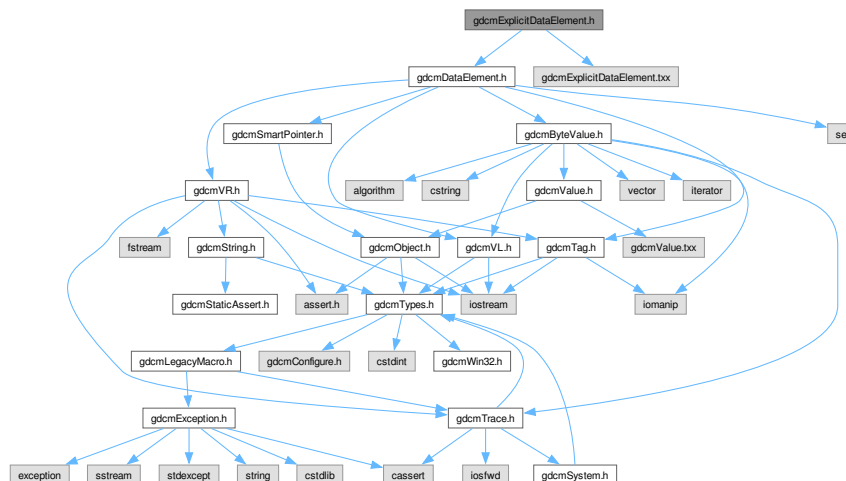
13.135 gdcnExplicitDataElement.h File Reference

```

#include "gdcnDataElement.h"
#include "gdcnExplicitDataElement.txx"

```


Include dependency graph for gdcmExplicitDataElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ExplicitDataElement](#)
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Namespaces

- namespace [gdcm](#)

13.136 gdcmExplicitDataElement.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
```


Classes

- class [gdcm::ExplicitImplicitDataElement](#)
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

Namespaces

- namespace [gdcm](#)

13.138 gdcmExplicitImplicitDataElement.h

[Go to the documentation of this file.](#)

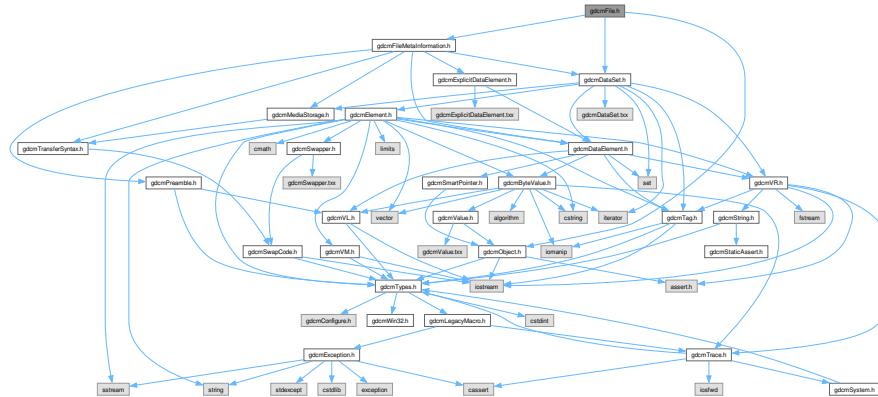
```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMEXPLICITIMPLICITDATAELEMENT_H
00015  #define GDCMEXPLICITIMPLICITDATAELEMENT_H
00016
00017  #include "gdcmDataElement.h"
00018
00019  namespace gdcm
00020  {
00021  // Data Element (ExplicitImplicit)
00022  class GDCM_EXPORT ExplicitImplicitDataElement : public DataElement
00023  {
00024  public:
00025  VL GetLength() const;
00026
00027  template <typename TSwap>
00028  std::istream &Read(std::istream &is);
00029
00030  template <typename TSwap>
00031  std::istream &ReadPreValue(std::istream &is);
00032
00033  template <typename TSwap>
00034  std::istream &ReadValue(std::istream &is, bool readvalues = true);
00035
00036  template <typename TSwap>
00037  std::istream &ReadWithLength(std::istream &is, VL & length)
00038  {
00039  (void)length;
00040  return Read<TSwap>(is);
00041  }
00042
00043  // PURPOSELY do not provide an implementation for writing !
00044  template <typename TSwap>
00045  //const std::ostream &Write(std::ostream &os) const;
00046  };
00047
00048  } // end namespace gdcm
00049
00050  #include "gdcmExplicitImplicitDataElement.txx"
00051
00052  #endif //GDCMEXPLICITIMPLICITDATAELEMENT_H

```

13.139 gdcmFile.h File Reference

```
#include "gdcmObject.h"
#include "gdcmDataSet.h"
#include "gdcmFileMetaInformation.h"
Include dependency graph for gdcmFile.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::File`
a DICOM File

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const File &val)`

13.140 gdcmFile.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMFILE_H
00015 #define GDCMFILE_H
00016
00017 #include "gdcmObject.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmFileMetaInformation.h"
00020
00021 namespace gdcm_ns
00022 {
00023
00024 class GDCM_EXPORT File : public Object
00025 {
00026 public:
00027     File();
00028     ~File() override;
00029
00030     friend std::ostream &operator«(std::ostream &os, const File &val);
00031
00032     std::istream &Read(std::istream &is);
00033
00034     std::ostream const &Write(std::ostream &os) const;
00035
00036     const FileMetaInformation &GetHeader() const { return Header; }
00037
00038     FileMetaInformation &GetHeader() { return Header; }
00039
00040     void SetHeader( const FileMetaInformation &fmi ) { Header = fmi; }
00041
00042     const DataSet &GetDataSet() const { return DS; }
00043
00044     DataSet &GetDataSet() { return DS; }
00045
00046     void SetDataSet( const DataSet &ds ) { DS = ds; }
00047
00048 private:
00049     FileMetaInformation Header;
00050     DataSet DS;
00051 };
00052
00053 //-----
00054 inline std::ostream& operator«(std::ostream &os, const File &val)
00055 {
00056     os « val.GetHeader() « std::endl;
00057     //os « val.GetDataSet() « std::endl; // FIXME
00058     gdcm_assert(0);
00059     return os;
00060 }
00061
00062 } // end namespace gdcm_ns
00063
00064 #endif //GDCMFILE_H

```

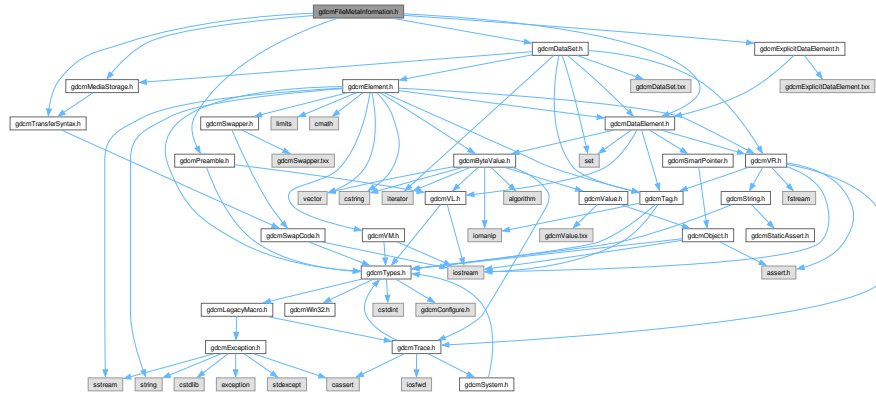
13.141 gdcmFileMetaInformation.h File Reference

```

#include "gdcmPreamble.h"
#include "gdcmDataSet.h"

```

```
#include "gdcmDataElement.h"
#include "gdcmMediaStorage.h"
#include "gdcmTransferSyntax.h"
#include "gdcmExplicitDataElement.h"
Include dependency graph for gdcmFileMetaInformation.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::FileMetaInformation`
Class to represent a `File` Meta Information.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const FileMetaInformation &val)`

13.142 gdcmFileMetaInformation.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMFILEMETAINFORMATION_H
00015 #define GDCMFILEMETAINFORMATION_H
00016
00017 #include "gdcmPreamble.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmDataElement.h"
00020 #include "gdcmMediaStorage.h"
00021 #include "gdcmTransferSyntax.h"
00022 #include "gdcmExplicitDataElement.h"
00023
00024 namespace gdcm_ns
00025 {
00040 class GDCM_EXPORT FileMetaInformation : public DataSet
00041 {
00042 public:
00043 // FIXME: TransferSyntax::TS_END -> TransferSyntax::ImplicitDataElement
00044 FileMetaInformation();
00045 ~FileMetaInformation();
00046
00047 friend std::ostream &operator<<(std::ostream &_os, const FileMetaInformation &_val);
00048
00049 bool IsValid() const { return true; }
00050
00051 TransferSyntax::NegociatedType GetMetaInformationTS() const { return MetaInformationTS; }
00052 void SetDataSetTransferSyntax(const TransferSyntax &ts);
00053 const TransferSyntax &GetDataSetTransferSyntax() const { return DataSetTS; }
00054 MediaStorage GetMediaStorage() const;
00055 std::string GetMediaStorageAsString() const;
00056
00057 // FIXME: no virtual function means: duplicate code...
00058 void Insert(const DataElement& de) {
00059     if( de.GetTag().GetGroup() == 0x0002 )
00060     {
00061         InsertDataElement( de );
00062     }
00063     else
00064     {
00065         gdcmErrorMacro( "Cannot add element with group != 0x0002 in the file meta header: " « de );
00066     }
00067 }
00068 void Replace(const DataElement& de) {
00069     Remove(de.GetTag());
00070     Insert(de);
00071 }
00072
00074 std::istream &Read(std::istream &is);
00075 std::istream &ReadCompat(std::istream &is);
00076
00078 std::ostream &Write(std::ostream &os) const;
00079
00081 void FillFromDataSet(DataSet const &ds);
00082
00084 const Preamble &GetPreamble() const { return P; }
00085 Preamble &GetPreamble() { return P; }
00086 void SetPreamble(const Preamble &p) { P = p; }
00087
00089 static void SetImplementationClassUID(const char * imp);
00090 static void AppendImplementationClassUID(const char * imp);
00091 static const char *GetImplementationClassUID();
00092 static void SetImplementationVersionName(const char * version);

```

```

00093 static const char *GetImplementationVersionName();
00094 static void SetSourceApplicationEntityTitle(const char * title);
00095 static const char *GetSourceApplicationEntityTitle();
00096
00097 FileMetaInformation(FileMetaInformation const& fmi) = default;
00098 FileMetaInformation& operator=(const FileMetaInformation& fmi) = default;
00099
00100 VL GetFullLength() const {
00101     return P.GetLength() + DataSet::GetLength<ExplicitDataElement>();
00102 }
00103
00104 protected:
00105 void ComputeDataSetTransferSyntax(); // FIXME
00106
00107 template <typename TSwap>
00108 std::istream &ReadCompatInternal(std::istream &is);
00109
00110 void Default();
00111 void ComputeDataSetMediaStorageSOPClass();
00112
00113 TransferSyntax DataSetTS;
00114 TransferSyntax::NegociatedType MetaInformationTS;
00115 MediaStorage::MSType DataSetMS;
00116
00117 protected:
00118 static const char * GetFileMetaInformationVersion();
00119 static const char * GetGDCMImplementationClassUID();
00120 static const char * GetGDCMImplementationVersionName();
00121 static const char * GetGDCMSourceApplicationEntityTitle();
00122
00123 private:
00124 Preamble P;
00125
00126 //static stuff:
00127 static const char GDCM_FILE_META_INFORMATION_VERSION[];
00128 static const char GDCM_IMPLEMENTATION_CLASS_UID[];
00129 static const char GDCM_IMPLEMENTATION_VERSION_NAME[];
00130 static const char GDCM_SOURCE_APPLICATION_ENTITY_TITLE[];
00131 static std::string ImplementationClassUID;
00132 static std::string ImplementationVersionName;
00133 static std::string SourceApplicationEntityTitle;
00134 };
00135 //-----
00136 inline std::ostream& operator<<(std::ostream &os, const FileMetaInformation &val)
00137 {
00138     os << val.GetPreamble() << std::endl;
00139     val.Print(os);
00140     return os;
00141 }
00142
00143 } // end namespace gdcm_ns
00144
00145 #endif //GDCMFILEMETAINFORMATION_H

```

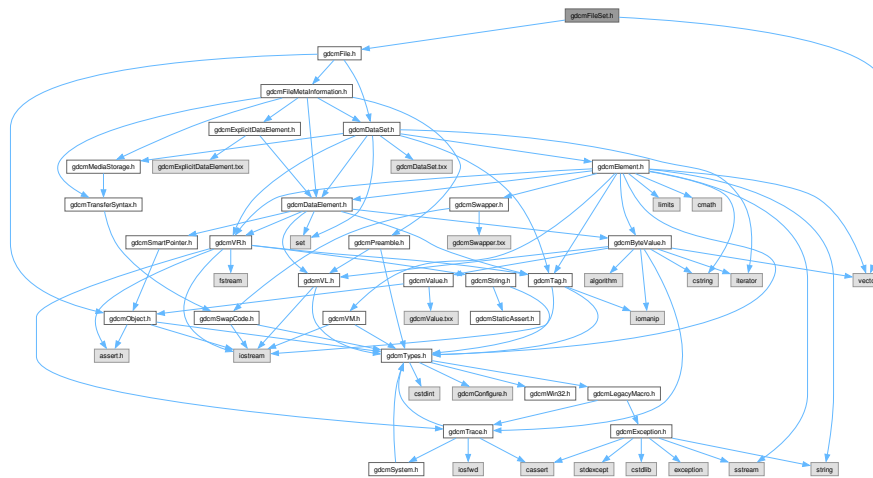
13.143 gdcmFileSet.h File Reference

```

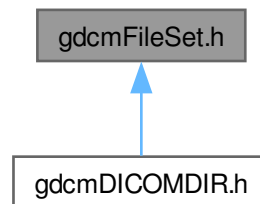
#include "gdcmFile.h"
#include <vector>

```


Include dependency graph for gdcmFileSet.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::FileSet](#)

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)`

13.144 gdcmFileSet.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMFILESET_H
00015  #define GDCMFILESET_H
00016  00017  #include "gdcmFile.h"
00018  #include <vector>
00019  00020  namespace gdcm
00021  {
00022  00026  class GDCM_EXPORT FileSet
00027  {
00028  friend std::ostream& operator<<(std::ostream &_os, const FileSet &d);
00029  public:
00030  FileSet():Files() {}
00031  typedef std::string FileType;
00032  typedef std::vector<FileType> FilesType;
00033  00035  void AddFile(File const & ) {}
00036  00039  bool AddFile(const char *filename);
00040  00041  void SetFiles(FilesType const &files);
00042  FilesType const &GetFiles() const {
00043  return Files;
00044  }
00045  00046  private:
00047  FilesType Files;
00048  };
00049  //-----
00050  inline std::ostream& operator<<(std::ostream &os, const FileSet &f)
00051  {
00052  (void)f; // FIXME
00053  return os;
00054  }
00055  00056  } // end namespace gdcm
00057  00058  #endif //GDCMFILESET_H

```

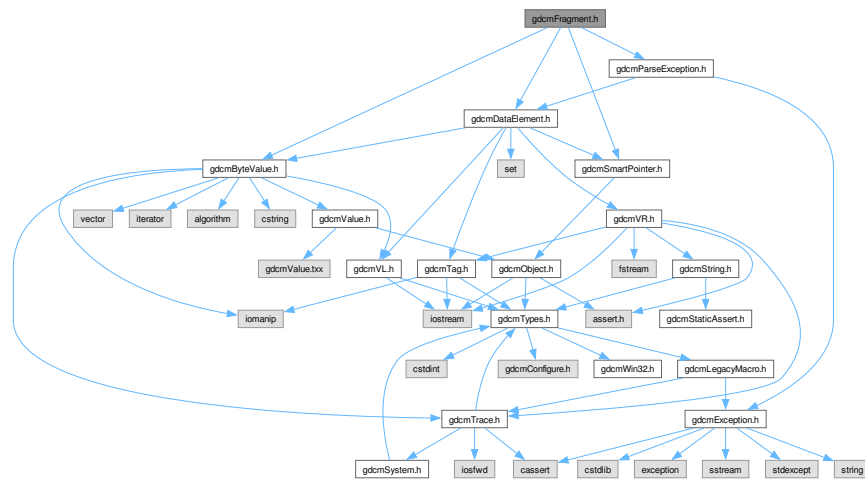
13.145 gdcmFragment.h File Reference

```

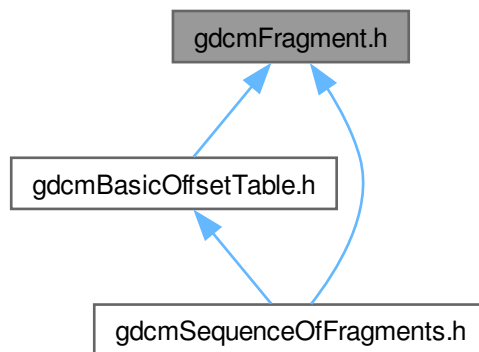
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"

```

Include dependency graph for gdcFragment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::Fragment`
Class to represent a `Fragment`.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)`

13.146 gdcmFragment.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMFRAGMENT_H
00015 #define GDCMFRAGMENT_H
00016
00017 #include "gdcmDataElement.h"
00018 #include "gdcmByteValue.h"
00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmParseException.h"
00021
00022 namespace gdcm_ns
00023 {
00024
00025  // Implementation detail:
00026  // I think Fragment should be a protected subclass of DataElement:
00027  // looking somewhat like this:
00028  /*
00029  class GDCM_EXPORT Fragment : protected DataElement
00030  {
00031  public:
00032      using DataElement::GetTag;
00033      using DataElement::GetVL;
00034      using DataElement::SetByteValue;
00035      using DataElement::GetByteValue;
00036      using DataElement::GetValue;
00037  */
00038  // Instead I am only hiding the SetTag member...
00039
00040  class GDCM_EXPORT Fragment : public DataElement
00041  {
00042  //protected:
00043  // void SetTag(const Tag &t);
00044  public:
00045      Fragment() : DataElement(Tag(0xfffe, 0xe000), 0) {}
00046      friend std::ostream &operator<<(std::ostream &os, const Fragment &val);
00047
00048      VL GetLength() const;
00049
00050      VL ComputeLength() const;
00051
00052      template <typename TSwap>
00053      std::istream &Read(std::istream &is)
00054      {
00055          ReadPreValue<TSwap>(is);
00056          return ReadValue<TSwap>(is);
00057      }
00058
00059      template <typename TSwap>
00060      std::istream &ReadPreValue(std::istream &is)
00061      {
00062          TagField.Read<TSwap>(is);
00063          if( !is )
00064          {
00065              // BogusItemStartItemEnd.dcm

```

```

00069     throw Exception( "Problem #1" );
00070 }
00071 if( !ValueLengthField.Read<TSwap>(is) )
00072 {
00073     // GENESIS_SIGNA-JPEG-CorruptFrag.dcm
00074     // JPEG fragment is declared to have 61902, but in fact really is only 61901
00075     // so we end up reading 0xddff,0x00e0, and VL = 0x0 (1 byte)
00076     throw Exception( "Problem #2" );
00077 }
00078 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00079 const Tag itemStart(0xfffe, 0xe000);
00080 const Tag seqDelItem(0xfffe, 0xe0dd);
00081 if( TagField != itemStart && TagField != seqDelItem )
00082 {
00083     throw Exception( "Problem #3" );
00084 }
00085 #endif
00086 return is;
00087 }
00088
00089 template <typename TSwap>
00090 std::istream &ReadValue(std::istream &is)
00091 {
00092     // Self
00093     SmartPointer<ByteValue> bv = new ByteValue;
00094     bv->SetLength(ValueLengthField);
00095     if( !bv->Read<TSwap>(is) )
00096     {
00097         // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
00098         gdcmWarningMacro( "Fragment could not be read" );
00099         //bv->SetLength(is.gcount());
00100         ValueField = bv;
00101         ParseException pe;
00102         pe.SetLastElement( *this );
00103         throw pe;
00104     }
00105     ValueField = bv;
00106     return is;
00107 }
00108
00109 template <typename TSwap>
00110 std::istream &ReadBacktrack(std::istream &is)
00111 {
00112     const Tag itemStart(0xfffe, 0xe000);
00113     const Tag seqDelItem(0xfffe, 0xe0dd);
00114
00115     bool cont = true;
00116     const std::streampos start = is.tellg();
00117     const int max = 10;
00118     int offset = 0;
00119     while( cont )
00120     {
00121         TagField.Read<TSwap>(is);
00122         gdcm_assert( is );
00123         if( TagField != itemStart && TagField != seqDelItem )
00124         {
00125             ++offset;
00126             is.seekg( (std::streampos)((size_t)start - offset) );
00127             gdcmWarningMacro( "Fuzzy Search, backtrack: " « (start - is.tellg()) « " Offset: " « is.tellg() );
00128             if( offset > max )
00129             {
00130                 gdcmErrorMacro( "Giving up" );
00131                 throw "Impossible to backtrack";
00132             }
00133         }
00134         else
00135         {
00136             cont = false;
00137         }
00138     }
00139     gdcm_assert( TagField == itemStart || TagField == seqDelItem );
00140     if( !ValueLengthField.Read<TSwap>(is) )
00141     {
00142         return is;
00143     }
00144
00145     // Self
00146     SmartPointer<ByteValue> bv = new ByteValue;
00147     bv->SetLength(ValueLengthField);
00148     if( !bv->Read<TSwap>(is) )
00149     {

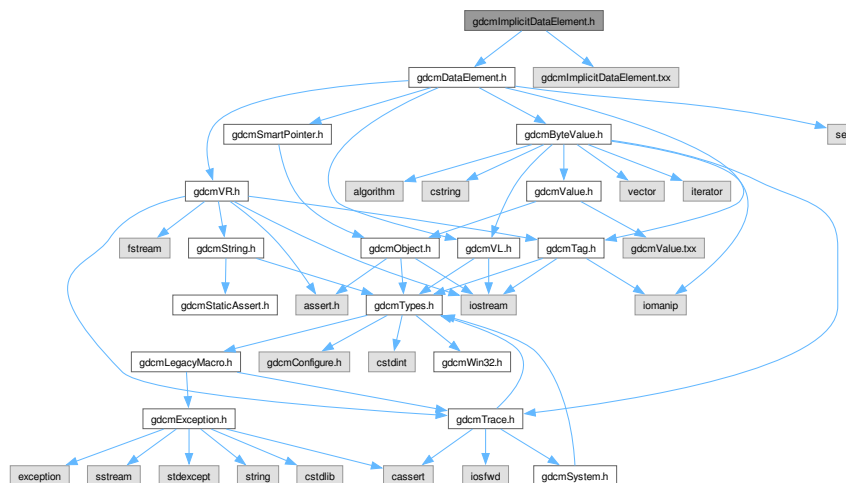
```

```

00150     // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
00151     gdcMWarningMacro( "Fragment could not be read" );
00152     //bv->SetLength(is.gcount());
00153     ValueField = bv;
00154     ParseException pe;
00155     pe.SetLastElement( *this );
00156     throw pe;
00157 }
00158 ValueField = bv;
00159 return is;
00160 }
00161
00162
00163 template <typename TSwap>
00164 std::ostream &Write(std::ostream &os) const {
00165     const Tag itemStart(0xfffe, 0xe000);
00166     const Tag seqDelItem(0xfffe, 0xe0dd);
00167     if( !TagField.Write<TSwap>(os) )
00168     {
00169         gdcM_assert(0 && "Should not happen");
00170         return os;
00171     }
00172     gdcM_assert( TagField == itemStart
00173         || TagField == seqDelItem );
00174     const ByteValue *bv = GetByteValue();
00175     // VL
00176     // The following piece of code is hard to read in order to support such broken file as:
00177     // CompressedLossy.dcm
00178     if( IsEmpty() )
00179     {
00180         //gdcM_assert( bv );
00181         VL zero = 0;
00182         if( !zero.Write<TSwap>(os) )
00183         {
00184             gdcM_assert(0 && "Should not happen");
00185             return os;
00186         }
00187     }
00188     else
00189     {
00190         gdcM_assert( ValueLengthField );
00191         gdcM_assert( !ValueLengthField.IsUndefined() );
00192         const VL actualLen = bv->ComputeLength();
00193         gdcM_assert( actualLen == ValueLengthField || actualLen == ValueLengthField + 1 );
00194         if( !actualLen.Write<TSwap>(os) )
00195         {
00196             gdcM_assert(0 && "Should not happen");
00197             return os;
00198         }
00199     }
00200     // Value
00201     if( ValueLengthField && bv )
00202     {
00203         // Self
00204         gdcM_assert( bv );
00205         gdcM_assert( bv->GetLength() == ValueLengthField );
00206         if( !bv->Write<TSwap>(os) )
00207         {
00208             gdcM_assert(0 && "Should not happen");
00209             return os;
00210         }
00211     }
00212     return os;
00213 }
00214 };
00215 //-----
00216 inline std::ostream &operator<<(std::ostream &os, const Fragment &val)
00217 {
00218     os << "Tag: " << val.TagField;
00219     os << "\tVL: " << val.ValueLengthField;
00220     if( val.ValueField )
00221     {
00222         os << "\t" << *(val.ValueField);
00223     }
00224
00225     return os;
00226 }
00227
00228 } // end namespace gdcM_ns
00229
00230 #endif //GDCMFRAGMENT_H

```

```
#include "gdcmDataElement.h"
#include "gdcmImplicitDataElement.txx"
Include dependency graph for gdcmImplicitDataElement.h:
```



- class `gdcmm::ImplicitDataElement`
Class to represent an Implicit VR Data Element.

- namespace `gdcm`

[Go to the documentation of this file.](#)

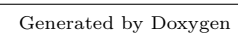
```

00001  /*=====
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*
00014 #ifndef GDCMIMPLICITDATAELEMENT_H

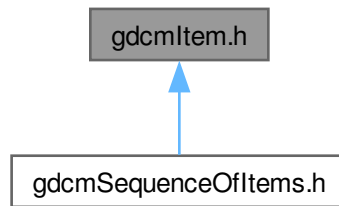
```

13.149 gdcmItem.h File Reference

Include dependency graph for gdcItem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Item`
Class to represent an `Item`.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)`

13.150 gdcmItem.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014
00015  #ifndef GDCMITEM_H
00016  #define GDCMITEM_H
00017
00018  #include "gdcmDataElement.h"
00019  #include "gdcmDataSet.h"
00020  #include "gdcmParseException.h"
00021  #include "gdcmSwapper.h"
  
```

```

00022
00023 #ifndef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00024 #include "gdcmByteSwapFilter.h"
00025 #endif
00026
00027 namespace gdcm_ns
00028 {
00029
00030 class DataSet;
00045 class GDCM_EXPORT Item : public DataElement
00046 {
00047 public:
00048 Item() : DataElement(Tag(0xfffe, 0xe000), 0xFFFFFFFF), NestedDataSet() {}
00049 friend std::ostream& operator<< (std::ostream &os, const Item &val);
00050
00051 void Clear() {
00052     this->DataElement::Clear();
00053     NestedDataSet.Clear();
00054 }
00055
00056 template <typename TDE>
00057 VL GetLength() const;
00058
00059 void InsertDataElement(const DataElement & de) {
00060     NestedDataSet.Insert(de);
00061     // Update the length
00062     if( !IsUndefinedLength() )
00063     {
00064         gdcm_assert( 0 && "InsertDataElement" );
00065         //ValueLengthField += de.GetLength();
00066     }
00067 }
00068 const DataElement& GetDataElement(const Tag& t) const
00069 {
00070     return NestedDataSet.GetDataElement(t);
00071 }
00072
00073 // Completely defines it with the nested dataset
00074 // destroy anything present
00075 void SetNestedDataSet(const DataSet& nested)
00076 {
00077     NestedDataSet = nested;
00078 }
00079 // Return a const ref to the Nested Data Set
00080 const DataSet &GetNestedDataSet() const
00081 {
00082     return NestedDataSet;
00083 }
00084 DataSet &GetNestedDataSet()
00085 {
00086     return NestedDataSet;
00087 }
00088
00089 //Value const & GetValue() const { return *NestedDataSet; }
00090
00091 Item(Item const &val):DataElement(val)
00092 {
00093     NestedDataSet = val.NestedDataSet;
00094 }
00095
00096 template <typename TDE, typename TSwap>
00097 std::istream &Read(std::istream &is) {
00098     // Superclass
00099     {
00100         DataSet &nested = NestedDataSet;
00101         nested.Clear();
00102         gdcm_assert( nested.IsEmpty() );
00103     }
00104     if( !TagField.Read<TSwap>(is) )
00105     {
00106         throw Exception("Should not happen (item)");
00107         return is;
00108     }
00109 #ifndef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00110     // MR_Philips_Intera_SwitchIndianess_noLgtSQItem_in_trueLgtSeq.dcm
00111     if( TagField == Tag(0xfeff, 0x00e0)
00112         || TagField == Tag(0xfeff, 0xdde0) )
00113     {
00114         gdcmWarningMacro( "ByteSwaping Private SQ: " « TagField );
00115         // Invert previously read TagField since wrong endianness:
00116         TagField = Tag( SwapperDoOp::Swap( TagField.GetGroup() ), SwapperDoOp::Swap( TagField.GetElement() ) );

```

```

00117     assert ( TagField == Tag(0xfffe, 0xe000)
00118     || TagField == Tag(0xfffe, 0xe0dd) );
00119
00120     if( !ValueLengthField.Read<SwapperDoOp>(is) )
00121     {
00122         gdcItem_assert(0 && "Should not happen");
00123         return is;
00124     }
00125     // Self
00126     // Some file written by GDCM 1.0 we write 0xFFFFFFFF instead of 0x0
00127     if( TagField == Tag(0xfffe, 0xe0dd) )
00128     {
00129         if( ValueLengthField )
00130         {
00131             gdcItemErrorMacro( "ValueLengthField is not 0" );
00132         }
00133     }
00134     //else if( ValueLengthField == 0 )
00135     // {
00136     //     //gdcItem_assert( TagField == Tag( 0xfffe, 0xe0dd) );
00137     //     if( TagField != Tag( 0xfffe, 0xe0dd) )
00138     //     {
00139     //         gdcItemErrorMacro( "SQ: " « TagField « " has a length of 0" );
00140     //     }
00141     // }
00142     else if( ValueLengthField.IsUndefined() )
00143     {
00144         DataSet &nested = NestedDataSet;
00145         nested.Clear();
00146         gdcItem_assert( nested.IsEmpty() );
00147         std::streampos start = is.tellg();
00148         try
00149         {
00150             nested.template ReadNested<TDE,SwapperDoOp>(is);
00151             ByteSwapFilter bsf(nested);
00152             bsf.ByteSwap();
00153         }
00154         catch(ParseException &pe)
00155         {
00156             (void)pe;
00157             // MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
00158             // You have to byteswap the length but not the tag...sigh
00159             gdcItemWarningMacro( "Attempt to read nested Item without byteswapping the Value Length." );
00160             start -= is.tellg();
00161             gdcItem_assert( start < 0 );
00162             is.seekg( start, std::ios::cur );
00163             nested.Clear();
00164             nested.template ReadNested<TDE,SwapperNoOp>(is);
00165             ByteSwapFilter bsf(nested);
00166             // Tag are read in big endian, need to byteswap them back...
00167             bsf.SetByteSwapTag(true);
00168             bsf.ByteSwap();
00169         }
00170         catch(Exception &e)
00171         {
00172             // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
00173             throw e;
00174         }
00175         catch(...)
00176         {
00177             gdcItem_assert(0);
00178         }
00179     }
00180     else /* if( ValueLengthField.IsUndefined() ) */
00181     {
00182         DataSet &nested = NestedDataSet;
00183         nested.Clear();
00184         gdcItem_assert( nested.IsEmpty() );
00185         nested.template ReadWithLength<TDE,SwapperDoOp>(is, ValueLengthField);
00186         ByteSwapFilter bsf(nested);
00187         bsf.ByteSwap();
00188     }
00189     return is;
00190 }
00191 // http://groups.google.com/group/comp.protocols.dicom/msg/c07efcf5e759fc83
00192 // Bug_Philips_ItemTag_3F3F.dcm
00193 if( TagField == Tag(0x3f3f, 0x3f00) )
00194 {
00195     //TagField = Tag(0xfffe, 0xe000);
00196 }
00197 #endif

```

```

00198 if( TagField != Tag(0xfffe, 0xe000) && TagField != Tag(0xfffe, 0xe0dd) )
00199 {
00200     gdcmlDebugMacro( "Invalid Item, found tag: " « TagField);
00201     throw Exception( "Not a valid Item" );
00202 }
00203 gdcml_assert( TagField == Tag(0xfffe, 0xe000) || TagField == Tag(0xfffe, 0xe0dd) );
00204
00205 if( !ValueLengthField.Read<TSwap>(is) )
00206 {
00207     gdcml_assert(0 && "Should not happen");
00208     return is;
00209 }
00210 // Self
00211 if( TagField == Tag(0xfffe, 0xe0dd) )
00212 {
00213     // Some file written by GDCM 1.0 were written with 0xFFFFFFFF instead of 0x0
00214     if( ValueLengthField )
00215     {
00216         gdcmlDebugMacro( "ValueLengthField is not 0 but " « ValueLengthField );
00217     }
00218 }
00219 else if( ValueLengthField.IsUndefined() )
00220 {
00221     DataSet &nested = NestedDataSet;
00222     nested.Clear();
00223     gdcml_assert( nested.IsEmpty() );
00224     nested.template ReadNested<TDE, TSwap>(is);
00225 }
00226 else /* if( ValueLengthField.IsUndefined() ) */
00227 {
00228     gdcml_assert( !ValueLengthField.IsUndefined() );
00229     DataSet &nested = NestedDataSet;
00230     nested.Clear();
00231     gdcml_assert( nested.IsEmpty() );
00232     nested.template ReadWithLength<TDE, TSwap>(is, ValueLengthField);
00233 }
00234
00235 return is;
00236 }
00237
00238 template <typename TDE, typename TSwap>
00239 const std::ostream &Write(std::ostream &os) const {
00240 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00241     if( TagField == Tag(0x3f3f, 0x3f00) && false )
00242     {
00243         Tag t(0xfffe, 0xe000);
00244         t.Write<TSwap>(os);
00245     }
00246     else
00247 #endif
00248     {
00249         assert ( TagField == Tag(0xfffe, 0xe000)
00250             || TagField == Tag(0xfffe, 0xe0dd) );
00251         // Not sure how this happen
00252         if( TagField == Tag(0xfffe, 0xe0dd) )
00253         {
00254             gdcmlWarningMacro( "SeqDelItem found in defined length Sequence" );
00255             gdcml_assert( ValueLengthField == 0 );
00256             gdcml_assert( NestedDataSet.Size() == 0 );
00257         }
00258         if( !TagField.Write<TSwap>(os) )
00259         {
00260             gdcml_assert(0 && "Should not happen");
00261             return os;
00262         }
00263     }
00264     if( ValueLengthField.IsUndefined() )
00265     {
00266         if( !ValueLengthField.Write<TSwap>(os) )
00267         {
00268             gdcml_assert(0 && "Should not happen");
00269             return os;
00270         }
00271     }
00272     else
00273     {
00274         const VL dummy = NestedDataSet.GetLength<TDE>();
00275         gdcml_assert( dummy % 2 == 0 );
00276         //gdcml_assert( ValueLengthField == dummy );
00277         if( !dummy.Write<TSwap>(os) )
00278         {

```

```

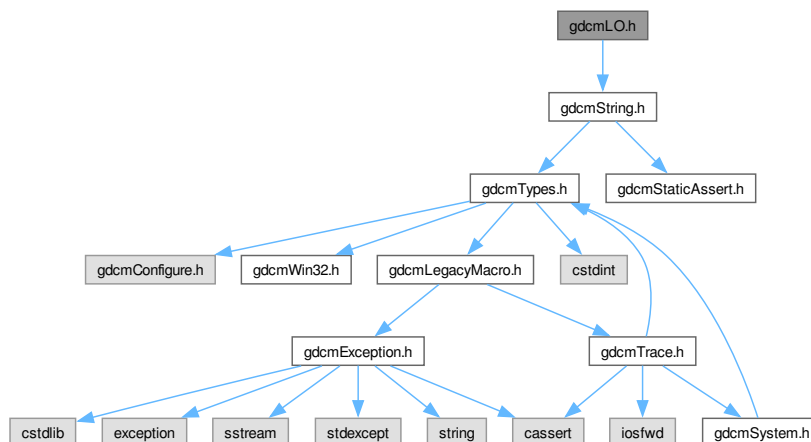
00279     gdcm_assert(0 && "Should not happen");
00280     return os;
00281 }
00282 }
00283 // Self
00284 NestedDataSet.Write<TDE,TSwap>(os);
00285 if( ValueLengthField.IsUndefined() )
00286 {
00287     const Tag itemDelItem(0xfffe,0xe00d);
00288     itemDelItem.Write<TSwap>(os);
00289     VL zero = 0;
00290     zero.Write<TSwap>(os);
00291 }
00292
00293 return os;
00294 }
00295
00296 /*
00297 There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed
00298 by the Transfer Syntax. They shall be encoded as Implicit VR. These special Data Elements are Item
00299 (FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD).
00300 However, the Data Set within the Value Field of the Data Element Item (FFFE,E000) shall be encoded
00301 according to the rules conveyed by the Transfer Syntax.
00302 */
00303 bool FindDataElement(const Tag &t) const {
00304     return NestedDataSet.FindDataElement( t );
00305 }
00306
00307 private:
00308 /* NESTED DATA SET  a Data Set contained within a Data Element of an other Data Set.
00309  * May be nested recursively.
00310  * Only Data Elements with VR = SQ may, themselves, contain Data Sets
00311  */
00312 DataSet NestedDataSet;
00313 };
00314 //-----
00315 inline std::ostream& operator<<(std::ostream& os, const Item &val)
00316 {
00317     os << val.TagField;
00318     os << "\t" << val.ValueLengthField << "\n";
00319     val.NestedDataSet.Print( os, "\t" );
00320
00321     return os;
00322 }
00323
00324
00325 } // end namespace gdcm_ns
00326
00327 #include "gdcmItem.txx"
00328
00329 #endif //GDCMITEM_H

```

13.151 gdcmLO.h File Reference

#include "gdcmString.h"

Include dependency graph for gdcmLO.h:



Classes

- class [gdcm::LO](#)
[LO](#).

Namespaces

- namespace [gdcm](#)

13.152 gdcmLO.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMLO_H
00015 #define GDCMLO_H
00016

```

```

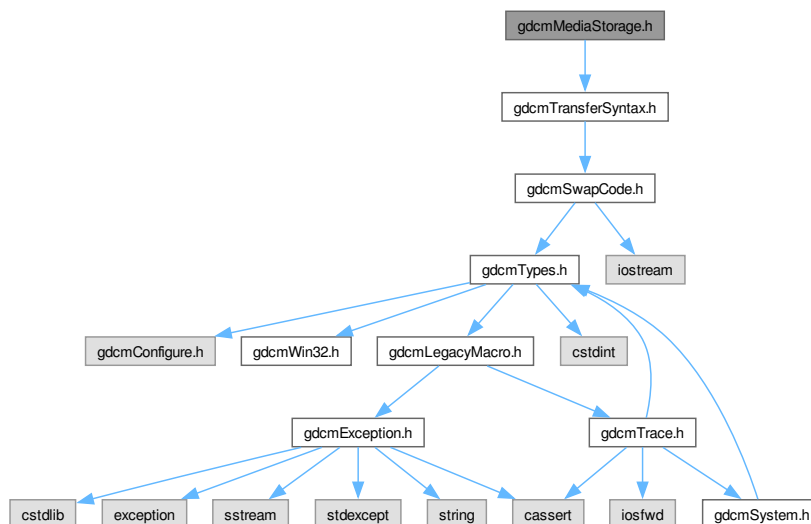
00017 #include "gdcmString.h"
00018
00019 namespace gdcm
00020 {
00021
00027 class /*GDCM_EXPORT*/ LO : public String<'\\',64> /* PLEASE do not export me */
00028 {
00029 public:
00030 // typedef are not inherited:
00031 typedef String<'\\',64> Superclass;
00032 typedef Superclass::value_type value_type;
00033 typedef Superclass::pointer pointer;
00034 typedef Superclass::reference reference;
00035 typedef Superclass::const_reference const_reference;
00036 typedef Superclass::size_type size_type;
00037 typedef Superclass::difference_type difference_type;
00038 typedef Superclass::iterator iterator;
00039 typedef Superclass::const_iterator const_iterator;
00040 typedef Superclass::reverse_iterator reverse_iterator;
00041 typedef Superclass::const_reverse_iterator const_reverse_iterator;
00042
00043 // LO constructors.
00044 LO(): Superclass() {}
00045 LO(const value_type* s): Superclass(s) {}
00046 LO(const value_type* s, size_type n): Superclass(s, n) {}
00047 LO(const Superclass& s, size_type pos=0, size_type n=npos):
00048     Superclass(s, pos, n) {}
00049
00050 bool IsValid() const {
00051     if( !Superclass::IsValid() ) return false;
00052     // Implementation specific:
00053     return true;
00054 }
00055 };
00056
00057 } // end namespace gdcm
00058
00059 #endif //GDCMLO_H

```

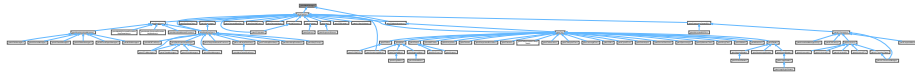
13.153 gdcmMediaStorage.h File Reference

#include "gdcmTransferSyntax.h"

Include dependency graph for gdcmMediaStorage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::MediaStorage](#)
[MediaStorage](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

13.154 gdcmMediaStorage.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMMEDIASTORAGE_H
00015  #define GDCMMEDIASTORAGE_H
00016
00017  #include "gdcmTransferSyntax.h"
00018
00019  namespace gdcm { class Tag; }
00020  namespace gdcm_ns
00021  {
00022  #if !defined(SWIGPYTHON) && !defined(SWIGSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023  using namespace gdcm;
00024  #endif
00025  class DataSet;
00026  class FileMetaInformation;
00027  class File;
00028
00029  // WARNING: This class will be deprecated in the future. There is no reason to extend this class.
00030  // Please check the gdcm::UIDs class if adding new well known UID.
00031
00043  class GDCM_EXPORT MediaStorage
00044  {
00045  public:
00046  typedef enum {
00047  MediaStorageDirectoryStorage = 0,

```


00048 ComputedRadiographyImageStorage,
00049 DigitalXRayImageStorageForPresentation,
00050 DigitalXRayImageStorageForProcessing,
00051 DigitalMammographyImageStorageForPresentation,
00052 DigitalMammographyImageStorageForProcessing,
00053 DigitalIntraoralXRayImageStorageForPresentation,
00054 DigitalIntraoralXRayImageStorageForProcessing,
00055 CTImageStorage,
00056 EnhancedCTImageStorage,
00057 UltrasoundImageStorageRetired,
00058 UltrasoundImageStorage,
00059 UltrasoundMultiFrameImageStorageRetired,
00060 UltrasoundMultiFrameImageStorage,
00061 MRIImageStorage,
00062 EnhancedMRIImageStorage,
00063 MRSpectroscopyStorage,
00064 NuclearMedicineImageStorageRetired,
00065 SecondaryCaptureImageStorage,
00066 MultiframeSingleBitSecondaryCaptureImageStorage,
00067 MultiframeGrayscaleByteSecondaryCaptureImageStorage,
00068 MultiframeGrayscaleWordSecondaryCaptureImageStorage,
00069 MultiframeTrueColorSecondaryCaptureImageStorage,
00070 StandaloneOverlayStorage,
00071 StandaloneCurveStorage,
00072 LeadECGWaveformStorage, // 12-
00073 GeneralECGWaveformStorage,
00074 AmbulatoryECGWaveformStorage,
00075 HemodynamicWaveformStorage,
00076 CardiacElectrophysiologyWaveformStorage,
00077 BasicVoiceAudioWaveformStorage,
00078 StandaloneModalityLUTStorage,
00079 StandaloneVOILUTStorage,
00080 GrayscaleSoftcopyPresentationStateStorageSOPClass,
00081 XRayAngiographicImageStorage,
00082 XRayRadiofluoroscopicImageStorage,
00083 XRayAngiographicBiPlaneImageStorageRetired,
00084 NuclearMedicineImageStorage,
00085 RawDataStorage,
00086 SpacialRegistrationStorage, // Spatial
00087 SpacialFiducialsStorage, // Spatial..
00088 PETImageStorage,
00089 RTImageStorage,
00090 RTDoseStorage,
00091 RTStructureSetStorage,
00092 RTPlanStorage,
00093 CSANonImageStorage,
00094 Philips3D,
00095 EnhancedSR,
00096 BasicTextSR,
00097 HardcopyGrayscaleImageStorage,
00098 ComprehensiveSR,
00099 DetachedStudyManagementSOPClass,
00100 EncapsulatedPDFStorage,
00101 EncapsulatedCDASTorage,
00102 StudyComponentManagementSOPClass,
00103 DetachedVisitManagementSOPClass,
00104 DetachedPatientManagementSOPClass,
00105 VideoEndoscopicImageStorage,
00106 GeneralElectricMagneticResonanceImageStorage,
00107 GEPrivate3DModelStorage,
00108 ToshibaPrivateDataStorage,
00109 MammographyCADSR,
00110 KeyObjectSelectionDocument,
00111 HangingProtocolStorage,
00112 ModalityPerformedProcedureStepSOPClass,
00113 PhilipsPrivateMRSyntheticImageStorage,
00114 VLPPhotographicImageStorage,
00115 SegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.4"
00116 RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
00117 XRay3DAngiographicImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.1
00118 EnhancedXAImageStorage,
00119 RTIonBeamsTreatmentRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.9
00120 SurfaceSegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.5"
00121 VLWholeSlideMicroscopyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.6
00122 RTTreatmentSummaryRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.7
00123 EnhancedUSVolumeStorage, // 1.2.840.10008.5.1.4.1.1.6.2
00124 XRayRadiationDoseSR, // 1.2.840.10008.5.1.4.1.1.88.67
00125 VLEndoscopicImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.1
00126 BreastTomosynthesisImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.3
00127 FujiPrivateCRImageStorage, // 1.2.392.200036.9125.1.1.2
00128 OphthalmicPhotography8BitImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.1

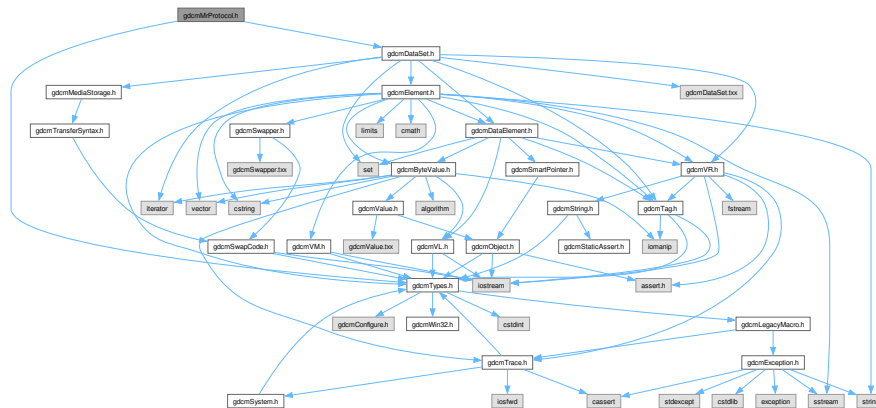
```

00129   OphthalmicTomographyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.4
00130   VLMicroscopicImageStorage,
00131   EnhancedPETImageStorage,
00132   VideoPhotographicImageStorage,
00133   XRay3DCraniofacialImageStorage,
00134   IVOCForPresentation,
00135   IVOCForProcessing,
00136   LegacyConvertedEnhancedCTImageStorage,
00137   LegacyConvertedEnhancedMRImageStorage,
00138   LegacyConvertedEnhancedPETImageStorage,
00139   BreastProjectionXRayImageStorageForPresentation,
00140   BreastProjectionXRayImageStorageForProcessing,
00141   HardcopyColorImageStorage,
00142   EnhancedMRColorImageStorage,
00143   FujiPrivateMammoCRImageStorage,
00144   OphthalmicPhotography16BitImageStorage,
00145   VideoMicroscopicImageStorage,
00146   MS_END
00147 } MStype; // Media Storage Type
00148
00149 typedef enum {
00150     NoObject = 0, // DICOMDIR
00151     Video, // Most common, include image, video and volume
00152     Waveform, // Isn't it simply a 1D video ?
00153     Audio, // ???
00154     PDF,
00155     URI, // URL...
00156     Segmentation, // TODO
00157     ObjectEnd
00158 } ObjectType;
00159
00161 static const char* GetMSString(MStype ts);
00162
00164 const char* GetString() const;
00165 static MStype GetMStype(const char *str);
00166
00167 MediaStorage(MStype type = MS_END):MSField(type) {}
00168
00171 static bool IsImage(MStype ts);
00172
00173 operator MStype () const { return MSField; }
00174
00175 const char *GetModality() const;
00176 unsigned int GetModalityDimension() const;
00177
00178 static unsigned int GetNumberOfMStype();
00179 static unsigned int GetNumberOfMSString();
00180 static unsigned int GetNumberOfModality();
00181
00182
00187 bool SetFromFile(File const &file);
00188
00191 bool SetFromDataSet(DataSet const &ds); // Will get the SOP Class UID
00192 bool SetFromHeader(FileMetaInformation const &fmi); // Will get the Media Storage SOP Class UID
00193 bool SetFromModality(DataSet const &ds);
00194 void GuessFromModality(const char *modality, unsigned int dimension = 2);
00195
00196 friend std::ostream &operator<<(std::ostream &os, const MediaStorage &ms);
00197
00198 bool IsUndefined() const { return MSField == MS_END; }
00199
00200 protected:
00201 void SetFromSourceImageSequence(DataSet const &ds);
00202
00203 private:
00204 bool SetFromDataSetOrHeader(DataSet const &ds, const Tag &tag);
00205
00206 std::string GetFromDataSetOrHeader(DataSet const &ds, const Tag &tag);
00207 std::string GetFromHeader(FileMetaInformation const &fmi);
00208 std::string GetFromDataSet(DataSet const &ds);
00209
00210 private:
00211 MStype MSField;
00212 };
00213 //-----
00214 inline std::ostream &operator<<(std::ostream &_os, const MediaStorage &ms)
00215 {
00216     const char *msstring = MediaStorage::GetMSString(ms);
00217     _os << (msstring ? msstring : "INVALID MEDIA STORAGE");
00218     return _os;
00219 }

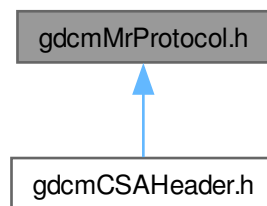
```

```
00220 }
00221
00222 } // end namespace gdcmm_ns
00223
00224 #endif // GDCMMEDIASTORAGE_H
```

```
#include "gdcTypes.h"
#include "gdcDataSet.h"
Include dependency graph for gdcMrProtocol.h:
```



This graph shows which files directly or indirectly include this file:



- class `gdcmm::MrProtocol`
Class for `MrProtocol`.
- struct `gdcmm::MrProtocol::Slice`
- struct `gdcmm::MrProtocol::SliceArray`
- struct `gdcmm::MrProtocol::Vector3`

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const MrProtocol &d)`

13.156 [gdcmMrProtocol.h](#)

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMMRPROTOCOL_H
00015  #define GDCMMRPROTOCOL_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmDataSet.h"
00019  00020  namespace gdcm
00021  {
00022  class ByteValue;
00023  /*
00024  * Everything done in this code is for the sole purpose of writing interoperable
00025  * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00026  * If you believe anything in this code violates any law or any of your rights,
00027  * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00028  * find a solution.
00029  */
00030  //-----
00031  00032  class DataElement;
00033  class GDCM_EXPORT MrProtocol
00034  {
00035  friend std::ostream& operator«(std::ostream &_os, const MrProtocol &d);
00036  public :
00037  MrProtocol();
00038  ~MrProtocol();
00039  00040  bool Load( const ByteValue * bv, const char * str, int version );
00041  void Print(std::ostream &os) const;
00042  00043  int GetVersion() const;
00044  const char * GetMrProtocolByName(const char *name) const;
00045  00046  bool FindMrProtocolByName(const char *name) const;
00047  00048  struct Vector3
00049  {
00050  double dSag;
00051  double dCor;
00052  double dTra;
00053  };
00054  struct Slice
00055  {
00056  Vector3 Normal;

```

```

00061     Vector3 Position;
00062 };
00063 struct SliceArray
00064 {
00065     std::vector< Slice > Slices;
00066 };
00067 bool GetSliceArray( MrProtocol::SliceArray & sa ) const;
00068
00069 private:
00070     struct Element;
00071     struct Internals;
00072     Internals *Pimpl;
00073 };
00074 //-----
00075 inline std::ostream& operator<<(std::ostream &os, const MrProtocol &d)
00076 {
00077     d.Print( os );
00078     return os;
00079 }
00080
00081 } // end namespace gdcm
00082 //-----
00083 #endif //GDCMMRPROTOCOL_H

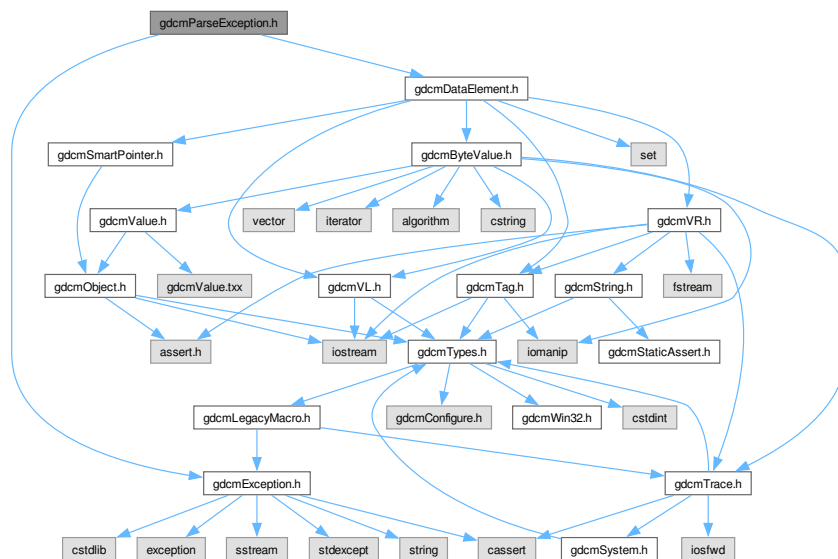
```

13.157 gdcmParseException.h File Reference

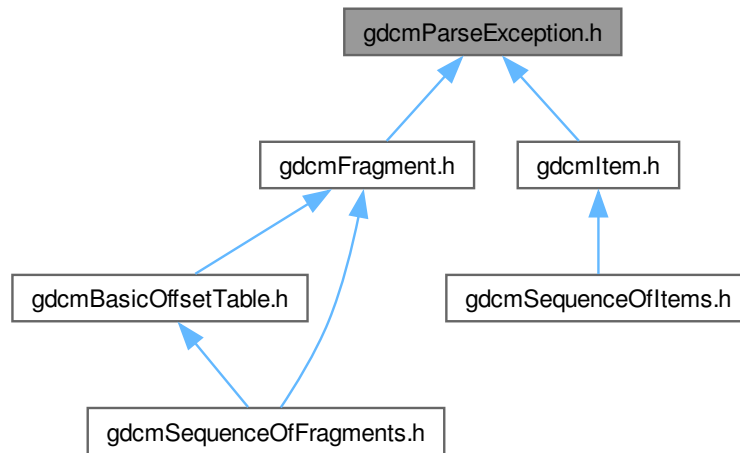
#include "gdcmException.h"

#include "gdcmDataElement.h"

Include dependency graph for gdcmParseException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ParseException](#)
[ParseException](#) Standard exception handling object.

Namespaces

- namespace [gdcm](#)

13.158 gdcmParseException.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPARSEEXCEPTION_H
00015  #define GDCMPARSEEXCEPTION_H
00016
00017  #include "gdcmException.h"
  
```

```

00018 #include "gdcmDataElement.h"
00019
00020 // Disable clang warning "dynamic exception specifications are deprecated".
00021 // We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
00022 // specifier we'll get an error in C++03 by not matching the superclass.
00023 #if defined(__clang__) && defined(__has_warning)
00024 # if __has_warning("-Wdeprecated")
00025 # pragma clang diagnostic push
00026 # pragma clang diagnostic ignored "-Wdeprecated"
00027 # endif
00028 #endif
00029
00030 namespace gdcm_ns
00031 {
00032 class ParseException : public Exception
00033 {
00034 public:
00035 ParseException() = default;
00036 ~ParseException() throw() override {}
00037
00038 ParseException &operator= ( const ParseException &orig )
00039 {
00040 LastElement = orig.LastElement;
00041 return *this;
00042 }
00043 ParseException(const ParseException& orig):Exception(orig)
00044 {
00045 LastElement = orig.LastElement;
00046 }
00047
00048 /* virtual bool operator==( const ParseException &orig )
00049 {
00050 return true;
00051 }*/
00052
00053 /*
00054 // Multiple calls to what ??
00055 const char* what() const throw()
00056 {
00057 static std::string strwhat;
00058 std::ostringstream oswhat;
00059 oswhat << "File " << " " << "Line " << ":\n";
00060 oswhat << "Description";
00061 strwhat = oswhat.str();
00062 return strwhat.c_str();
00063 }
00064 */
00065 void SetLastElement(const DataElement& de)
00066 {
00067 LastElement = de;
00068 }
00069 const DataElement& GetLastElement() const { return LastElement; }
00070
00071 private:
00072 // Store last parsed element before error:
00073 DataElement LastElement;
00074 };
00075
00076 } // end namespace gdcm_ns
00077
00078 // Undo warning suppression.
00079 #if defined(__clang__) && defined(__has_warning)
00080 # if __has_warning("-Wdeprecated")
00081 # pragma clang diagnostic pop
00082 # endif
00083 #endif
00084
00085 #endif

```

13.159 gdcmParser.h File Reference

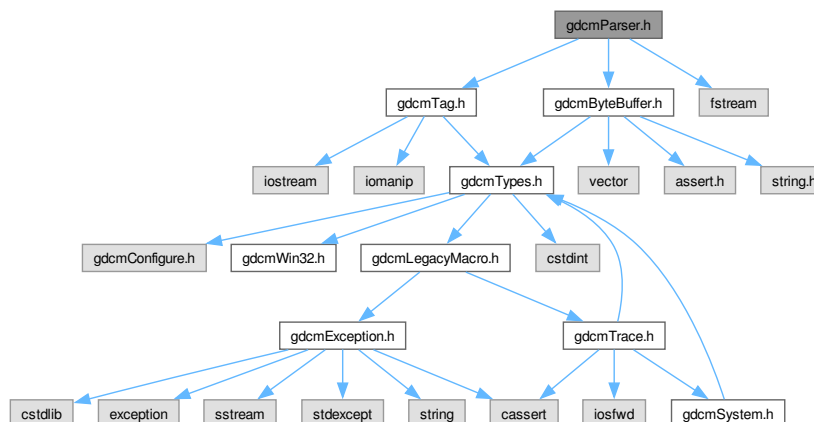
```

#include "gdcmTag.h"
#include "gdcmByteBuffer.h"

```

```
#include <fstream>
```

Include dependency graph for gdcParser.h:



Classes

- class [gdcm::Parser](#)
[Parser](#) ala XML_Parser from expat (SAX).

Namespaces

- namespace [gdc](#)

13.160 gdcParser.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  00015  #ifndef GDCMPARSER_H
00016  #define GDCMPARSER_H
00017  00018  #include "gdcTag.h"
00019  #error do not use
00020  #include "gdcByteBuffer.h"
00021

```



```

00022 #include <fstream> // std::ifstream
00023
00024 namespace gdcM
00025 {
00032 class GDCM_EXPORT Parser /*: private IStream*/
00033 {
00034 public:
00035     typedef enum {
00036         NoError,
00037         NoMemoryError,
00038         SyntaxError,
00039         NoElementsError,
00040         TagMismatchError,
00041         DuplicateAttributeError,
00042         JunkAfterDocElementError,
00043         UndefinedEntityError,
00044         UnexpectedStateError
00045     } ErrorType;
00046
00047     Parser() : UserData(0), Buffer(), ErrorCode(NoError) {}
00048     ~Parser() {}
00049
00050     // Parse some more of the document. The string s is a buffer containing
00051     // part (or perhaps all) of the document. The number of bytes of s that
00052     // are part of the document is indicated by len. This means that s
00053     // doesn't have to be null terminated. It also means that if len is
00054     // larger than the number of bytes in the block of memory that s points
00055     // at, then a memory fault is likely. The isFinal parameter informs the
00056     // parser that this is the last piece of the document. Frequently, the
00057     // last piece is empty (i.e. len is zero.) If a parse error occurred,
00058     // it returns 0. Otherwise it returns a non-zero value.
00059     bool Parse(const char* s, int len, bool isFinal);
00060
00061     // Set handlers for start and end tags. Attributes are passed to the
00062     // start handler as a pointer to a vector of char pointers. Each
00063     // attribute seen in a start (or empty) tag occupies 2 consecutive places
00064     // in this vector: the attribute name followed by the attribute value.
00065     // These pairs are terminated by a null pointer.
00066     typedef void (*StartElementHandler) (void *userData,
00067                                         const Tag &tag,
00068                                         const char *atts[]);
00069     typedef void (*EndElementHandler) (void *userData, const Tag &name);
00070     void SetElementHandler(StartElementHandler start, EndElementHandler end);
00071
00072     // Return what type of error has occurred.
00073     ErrorType GetErrorCode() const;
00074
00075     // Return a string describing the error corresponding to code.
00076     // The code should be one of the enums that can be returned from
00077     // GetErrorCode.
00078     static const char *GetErrorString(ErrorType const &err);
00079
00080     // Return the byte offset of the position.
00081     unsigned long GetCurrentByteIndex() const;
00082
00083     // Miscellaneous functions
00084
00085     // The functions in this section either obtain state information from
00086     // the parser or can be used to dynamically set parser options.
00087
00088     // This sets the user data pointer that gets passed to handlers.
00089     void SetUserData(void *userData);
00090
00091     // This returns the user data pointer that gets passed to handlers.
00092     void * GetUserData() const;
00093
00094 protected:
00095
00096     // This is just like Parse, except in this case expat provides the buffer.
00097     // By obtaining the buffer from expat with the GetBuffer function,
00098     // the application can avoid double copying of the input.
00099     bool ParseBuffer(int len, bool isFinal);
00100
00101     // Obtain a buffer of size len to read a piece of the document into.
00102     // A NULL value is returned if expat can't allocate enough memory for
00103     // this buffer. This has to be called prior to every call to ParseBuffer.
00104     char *GetBuffer(int len);
00105
00106     ErrorType Process();
00107
00108 private:

```

```

00109  std::ifstream Stream;
00110  void* UserData;
00111  ByteBuffer Buffer;
00112  ErrorType ErrorCode;
00113
00114  StartElementHandler StartElement;
00115  EndElementHandler EndElement;
00116  };
00117
00118  } // end namespace gdcm
00119
00120  #endif //GDCMPARSER_H

```

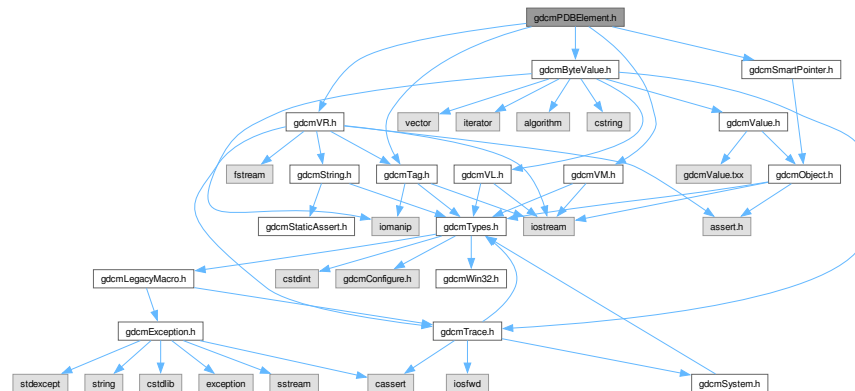
13.161 gdcmPDBelement.h File Reference

```

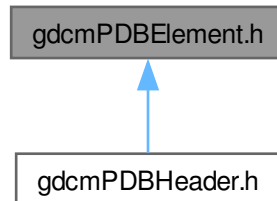
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for gdcmPDBelement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::PDBelement](#)
Class to represent a PDB [Element](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBelement &val)`

13.162 gdcmPDBelement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMPDBelement_H
00015  #define GDCMPDBelement_H
00016  00017  #include "gdcmTag.h"
00018  #include "gdcmVM.h"
00019  #include "gdcmVR.h"
00020  #include "gdcmByteValue.h"
00021  #include "gdcmSmartPointer.h"
00022  00023  namespace gdcm
00024  {
00025  00026  class GDCM_EXPORT PDBelement
00027  {
00028  public:
00029  PDBelement() = default;
00030  00031  friend std::ostream& operator<<(std::ostream &os, const PDBelement &val);
00032  00033  const char *GetName() const { return NameField.c_str(); }
00034  void SetName(const char *name) { NameField = name; }
00035  00036  const char *GetValue() const { return ValueField.c_str(); }
00037  void SetValue(const char *value) { ValueField = value; }
00038  00039  bool operator==(const PDBelement &de) const
00040  {
00041  return ValueField == de.ValueField
00042  && NameField == de.NameField;
00043  }
00044  00045  protected:
00046  std::string NameField;
00047  std::string ValueField;
00048  };
00049  //-----

```


13.164 gdcmPDBHeader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPDBHEADER_H
00015 #define GDCMPDBHEADER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmPDBelement.h"
00020
00021 namespace gdcm
00022 {
00023
00024  /*
00025   * Everything done in this code is for the sole purpose of writing interoperable
00026   * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00027   * If you believe anything in this code violates any law or any of your rights,
00028   * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00029   * find a solution.
00030   */
00031  //-----
00032
00033  class DataElement;
00034  class PrivateTag;
00035  class GDCM_EXPORT PDBHeader
00036  {
00037  friend std::ostream& operator<<(std::ostream &_os, const PDBHeader &d);
00038  public :
00039    PDBHeader() = default;
00040    ~PDBHeader() = default;
00041
00042    bool LoadFromDataElement(DataElement const &de);
00043
00044    void Print(std::ostream &os) const;
00045
00046    static const PrivateTag & GetPDBInfoTag();
00047
00048    const PDBelement &GetPDBelementByName(const char *name);
00049
00050    bool FindPDBelementByName(const char *name);
00051
00052  protected:
00053    const PDBelement& GetPDBeEnd() const;
00054
00055  private:
00056    int readprotocoldatablock(const char *input, size_t inputlen, bool verbose);
00057    std::vector<PDBelement> InternalPDBDataSet;
00058    static PDBelement PDBeEnd;
00059    bool IsXML;
00060    std::string xmltxt;
00061  };
00062  //-----
00063  inline std::ostream& operator<<(std::ostream &os, const PDBHeader &d)
00064  {
00065    d.Print( os );
00066    return os;
00067  }
00068  } // end namespace gdcm
00069  //-----
00070  #endif //GDCMPDBHEADER_H

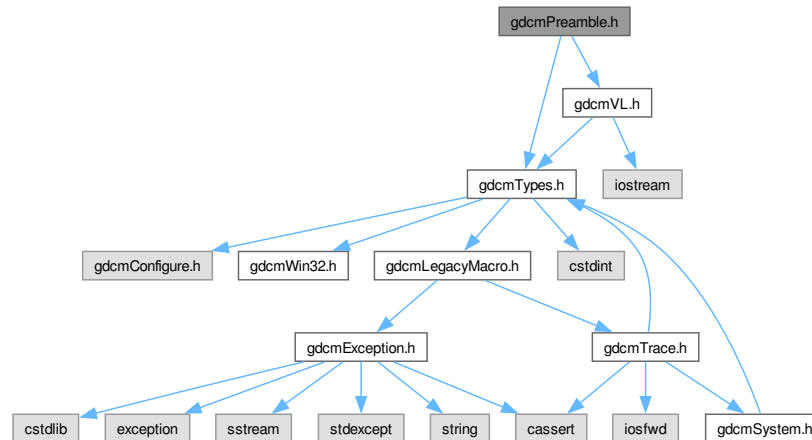
```

13.165 gdcmPreamble.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmVL.h"
```

Include dependency graph for gdcmPreamble.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Preamble](#)
DICOM [Preamble](#) (Part 10).

Namespaces

- namespace [gdcm](#)

Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [Preamble](#) &val)

13.166 gdcmPreamble.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPREAMBLE_H
00015 #define GDCMPREAMBLE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmVL.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class GDCM_EXPORT Preamble
00024 {
00025 public:
00026 Preamble();
00027 ~Preamble();
00028
00029 friend std::ostream &operator<<(std::ostream &_os, const Preamble &_val);
00030
00031 void Clear();
00032
00033 void Valid();
00034 void Create();
00035 void Remove();
00036
00037 std::istream &Read(std::istream &is);
00038
00039 std::ostream const &Write(std::ostream &os) const;
00040
00041 void Print(std::ostream &os) const;
00042
00043 const char *GetInternal() const { return Internal; }
00044
00045 bool IsEmpty() const { return !Internal; }
00046
00047 VL GetLength() const { return 128 + 4; }
00048
00049 Preamble(Preamble const &):Internal(nullptr)
00050 {
00051 Create();
00052 }
00053 Preamble& operator=(Preamble const &)
00054 {
00055 Create();
00056 return *this;
00057 }
00058 protected:
00059 //
00060 bool IsValid() const {
00061 // is (IsValid == true) => Internal was read
00062 return true;
00063 }
00064
00065 private:
00066 char *Internal;
00067 };
00068 //-----
00069 inline std::ostream& operator<<(std::ostream &os, const Preamble &val)
00070 {
00071 os << val.Internal;
00072 return os;
00073 }

```

```

00085
00086 } // end namespace gdcM
00087
00088 #endif //GDCMPREAMBLE_H

```

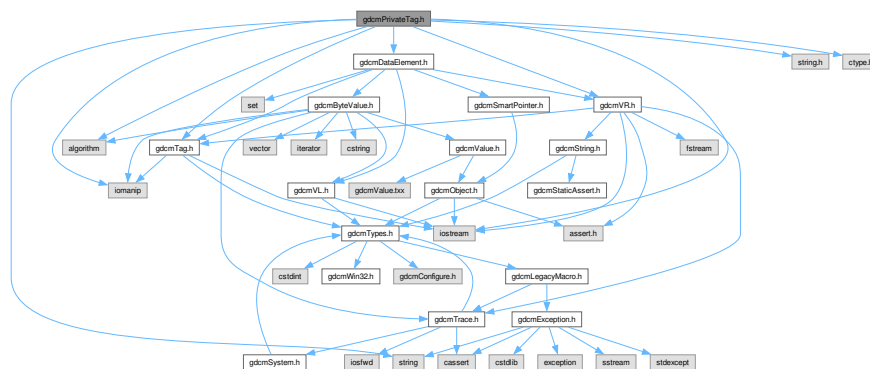
13.167 gdcMPrivateTag.h File Reference

```

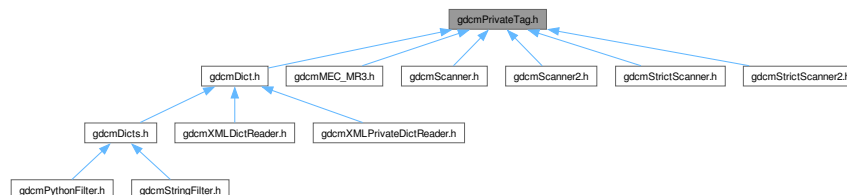
#include "gdcMTag.h"
#include "gdcMVR.h"
#include "gdcMDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>

```

Include dependency graph for gdcMPrivateTag.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcM::PrivateTag](#)

Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

13.168 gdcmPrivateTag.h

[Go to the documentation of this file.](#)

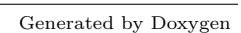
```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPRIVATETAG_H
00015 #define GDCMPRIVATETAG_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmVR.h"
00019 #include "gdcmDataElement.h"
00020
00021 #include <iostream>
00022 #include <iomanip>
00023 #include <string>
00024 #include <algorithm>
00025
00026 #include <string.h> // strlen
00027 #include <ctype.h> // tolower
00028
00029 namespace gdcm_ns
00030 {
00031
00032
00033 // TODO: We could save some space since we only store 8bits for element
00034 class GDCM_EXPORT PrivateTag : public Tag
00035 {
00036
00037     friend std::ostream& operator<<(std::ostream &_os, const PrivateTag &_val);
00038 public:
00039     PrivateTag(uint16_t group = 0, uint16_t element = 0, const char *owner = "") : Tag(group, element), Owner(owner ?
00040     LOComp::Trim(owner) : "") {
00041         // truncate the high bits
00042         SetElement( (uint8_t)element );
00043     }
00044     PrivateTag( Tag const & t, const char *owner = "") : Tag(t), Owner(owner ? LOComp::Trim(owner) : "") {
00045         // truncate the high bits
00046         SetElement( (uint8_t)t.GetElement());
00047     }
00048
00049     const char *GetOwner() const { return Owner.c_str(); }
00050     void SetOwner(const char *owner) { if(owner) Owner = LOComp::Trim(owner); }
00051
00052     PrivateTag &operator=(const PrivateTag &_val)
00053     {
00054         SetElementTag( _val.GetElementTag() );
00055         Owner = _val.Owner;
00056         return *this;
00057     }
00058
00059 }
00060

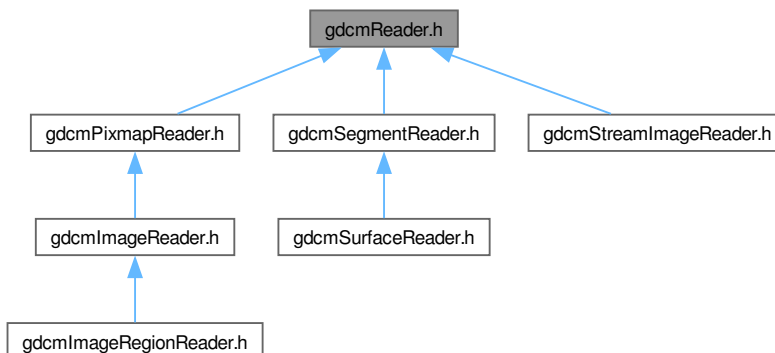
```

13.169 gdcmReader.h File Reference

Include dependency graph for `gdcmReader.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Reader](#)
[Reader](#) ala DOM (Document [Object](#) Model).

Namespaces

- namespace [gdcm](#)

13.170 gdcmReader.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMREADER_H
00015  #define GDCMREADER_H
00016  00017  #include "gdcmFile.h"
00018  00019  #include <fstream>
00020  00021  namespace gdcm_ns
00022  {
00023    class StreamImageReader;
00053    class GDCM_EXPORT Reader

```

```

00054 {
00055 public:
00056     Reader();
00057     virtual ~Reader();
00058
00060     virtual bool Read(); // Execute()
00061
00064     void SetFileName(const char *filename_native);
00065
00067     void SetStream(std::istream &input_stream) {
00068         Stream = &input_stream;
00069     }
00070
00072     const File &GetFile() const { return *F; }
00073
00075     File &GetFile() { return *F; }
00076
00078     void SetFile(File& file) { F = &file; }
00079
00082     bool ReadUpToTag(const Tag & tag, std::set<Tag> const & skiptags = std::set<Tag>() );
00083
00085     bool ReadSelectedTags(std::set<Tag> const & tags, bool readvalues = true);
00086
00088     bool ReadSelectedPrivateTags(std::set<PrivateTag> const & ptags, bool readvalues = true);
00089
00092     bool CanRead() const;
00093
00096     size_t GetStreamCurrentPosition() const;
00097
00098 protected:
00099     bool ReadPreamble();
00100     bool ReadMetaInformation();
00101     bool ReadDataSet();
00102
00103     SmartPointer<File> F;
00104
00105     friend class StreamImageReader; //need to be friended to be able to grab the GetStreamPtr
00106
00107     //this function is added for the StreamImageReader, which needs to read
00108     //up to the pixel data and then stops right before reading the pixel data.
00109     //it's used to get that position, so that reading can continue
00110     //apace once the read function is called.
00111     //so, this function gets the stream directly, and then allows for position information
00112     //from the tellg function, and allows for stream/pointer manip in order
00113     //to read the pixel data. Note, of course, that reading pixel elements
00114     //will still have to be subject to endianness swaps, if necessary.
00115     std::istream* GetStreamPtr() const { return Stream; }
00116
00117 private:
00118     template <typename T_Caller>
00119     bool InternalReadCommon(const T_Caller &caller);
00120     TransferSyntax GuessTransferSyntax();
00121     std::istream *Stream;
00122     std::ifstream *Ifstream;
00123
00124     // prevent copy/move to avoid 2 ifstream leak
00125     Reader(const Reader &) = delete;
00126     Reader &operator=(const Reader &) = delete;
00127     Reader(const Reader &&) = delete;
00128     Reader &operator=(const Reader &&) = delete;
00129 };
00130
00136
00137 } // end namespace gdcn_ns
00138
00139
00140 #endif //GDCMREADER_H

```

13.171 gdcnSequenceOfFragments.h File Reference

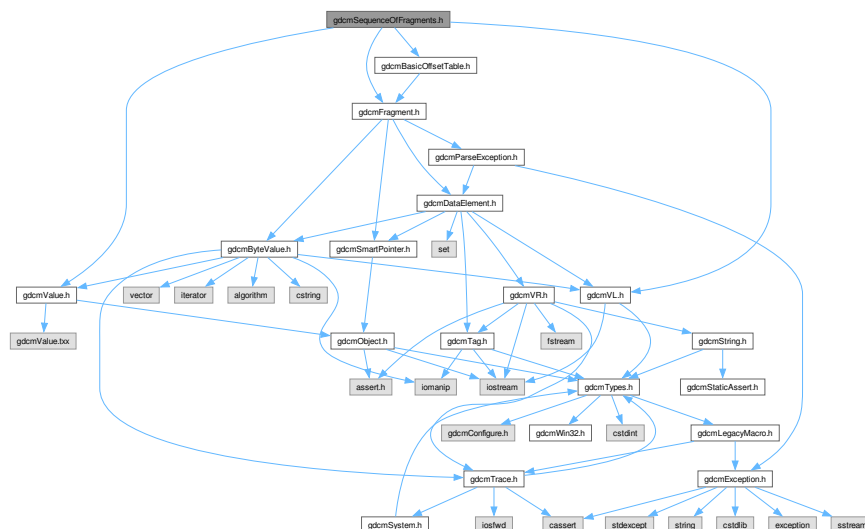
```

#include "gdcnValue.h"
#include "gdcnVL.h"
#include "gdcnFragment.h"

```

```
#include "gdcmBasicOffsetTable.h"
```

Include dependency graph for gdcmSequenceOfFragments.h:



Classes

- class `gdcm::SequenceOfFragments`
Class to represent a Sequence Of Fragments.

Namespaces

- namespace `gdcm`

13.172 gdcmSequenceOfFragments.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSEQUENCEOFFRAGMENTS_H
00015 #define GDCMSEQUENCEOFFRAGMENTS_H
00016
00017 #include "gdcmValue.h"
00018 #include "gdcmVL.h"
```

```

00019 #include "gdcmFragment.h"
00020 #include "gdcmBasicOffsetTable.h"
00021
00022 namespace gdcm_ns
00023 {
00024
00025 // FIXME gdcmSequenceOfItems and gdcmSequenceOfFragments
00026 // should be rethink (duplicate code)
00031 class GDCM_EXPORT SequenceOfFragments : public Value
00032 {
00033 public:
00034 // Typdefs:
00035 typedef std::vector<Fragment> FragmentVector;
00036 typedef FragmentVector::size_type SizeType;
00037 typedef FragmentVector::iterator Iterator;
00038 typedef FragmentVector::const_iterator ConstIterator;
00039 Iterator Begin() { return Fragments.begin(); }
00040 Iterator End() { return Fragments.end(); }
00041 ConstIterator Begin() const { return Fragments.begin(); }
00042 ConstIterator End() const { return Fragments.end(); }
00043
00045 SequenceOfFragments():Table(),SequenceLengthField(0xFFFFFFFF) { }
00046
00048 VL GetLength() const override {
00049     return SequenceLengthField;
00050 }
00051
00053 void SetLength(VL length) override {
00054     SequenceLengthField = length;
00055 }
00056
00058 void Clear() override;
00059
00061 void AddFragment(Fragment const &item);
00062
00063 // Compute the length of all fragments (and fragments only!).
00064 // Basically the size of the PixelData as stored (in bytes).
00065 unsigned long ComputeByteLength() const;
00066
00067 // Compute the length of fragments (in bytes)+ length of tag...
00068 // to be used for computation of Group Length
00069 VL ComputeLength() const;
00070
00071 // Get the buffer
00072 bool GetBuffer(char *buffer, unsigned long length) const;
00073 bool GetFragBuffer(unsigned int fragNb, char *buffer, unsigned long &length) const;
00074
00075 SizeType GetNumberOfFragments() const;
00076 const Fragment& GetFragment(SizeType num) const;
00077
00078 // Write the buffer of each fragment (call WriteBuffer on all Fragments, which are
00079 // ByteValue). No Table information is written.
00080 bool WriteBuffer(std::ostream &os) const;
00081
00082 const BasicOffsetTable &GetTable() const { return Table; }
00083 BasicOffsetTable &GetTable() { return Table; }
00084
00085 template <typename TSwap>
00086 std::istream& Read(std::istream &is, bool readvalues = true)
00087 {
00088     gdcm_assert( SequenceLengthField.IsUndefined() );
00089     ReadPreValue<TSwap>(is);
00090     return ReadValue<TSwap>(is, readvalues);
00091 }
00092
00093 template <typename TSwap>
00094 std::istream& ReadPreValue(std::istream &is)
00095 {
00096     // First item is the basic offset table:
00097     #if 0
00098     try
00099     {
00100         Table.Read<TSwap>(is);
00101         gdcmDebugMacro( "Table: " « Table );
00102     }
00103     catch(...)
00104     {
00105         // throw "SIEMENS Icon thingy";
00106         // Bug_Siemens_PrivateIconNoItem.dcm
00107         // First thing first let's rewind
00108         is.seekg(-4, std::ios::cur);

```

```

00109 // FF D8 <=> Start of Image (SOI) marker
00110 // FF E0 <=> APP0 Reserved for Application Use
00111 if ( Table.GetTag() == Tag(0xd8ff,0xe0ff) )
00112 {
00113     Table = BasicOffsetTable(); // clear up stuff
00114     //Table.SetByteValue( "", 0 );
00115     Fragment frag;
00116     if( FillFragmentWithJPEG( frag, is ) )
00117     {
00118         Fragments.push_back( frag );
00119     }
00120     return is;
00121 }
00122 else
00123 {
00124     throw "Catch me if you can";
00125     //gdcassert(0);
00126 }
00127 }
00128 #else
00129 Table.Read<TSwap>(is);
00130 gdcDebugMacro( "Table: " « Table );
00131 #endif
00132 return is;
00133 }
00134
00135 template <typename TSwap>
00136 std::istream& ReadValue(std::istream &is, bool /*readvalues*/)
00137 {
00138     const Tag seqDelItem(0xfffe,0xe0dd);
00139     // not used for now...
00140     Fragment frag;
00141     try
00142     {
00143         while( frag.Read<TSwap>(is) && frag.GetTag() != seqDelItem )
00144         {
00145             //gdcDebugMacro( "Frag: " « frag );
00146             Fragments.push_back( frag );
00147         }
00148         gdcassert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00149     }
00150     catch(Exception &ex)
00151     {
00152         (void)ex;
00153     }
00154 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00155     // that's ok ! In all cases the whole file was read, because
00156     // Fragment::Read only fail on eof() reached 1.
00157     // SIEMENS-JPEG-CorruptFrag.dcm is more difficult to deal with, we have a
00158     // partial fragment, read we decide to add it anyway to the stack of
00159     // fragments (eof was reached so we need to clear error bit)
00160     if( frag.GetTag() == Tag(0xfffe,0xe000) )
00161     {
00162         gdcWarningMacro( "Pixel Data Fragment could be corrupted. Use file at own risk" );
00163         Fragments.push_back( frag );
00164         is.clear(); // clear the error bit
00165     }
00166     // 2. GENESIS_SIGNA-JPEG-CorruptFrag.dcm
00167     else if ( frag.GetTag() == Tag(0xddff,0xe0e0) )
00168     {
00169         gdcassert( Fragments.size() == 1 );
00170         const ByteValue *bv = Fragments[0].GetByteValue();
00171         gdcassert( (unsigned char)bv->GetPointer()[ bv->GetLength() - 1 ] == 0xfe );
00172         // Yes this is an extra copy, this is a bug anyway, go fix YOUR code
00173         Fragments[0].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
00174         gdcWarningMacro( "JPEG Fragment length was declared with an extra byte"
00175             " at the end: stripped !" );
00176         is.clear(); // clear the error bit
00177     }
00178     // 3. LEICA/WSI
00179     else if ( (frag.GetTag().GetGroup() == 0x00ff)
00180         && ((frag.GetTag().GetElement() & 0x00ff) == 0xe0) )
00181     {
00182         // Looks like there is a mess with offset and odd byte array
00183         // We are going first to backtrack one byte back, and then use a
00184         // ReadBacktrack function which in turn may backtrack up to 10 bytes
00185         // backward. This appears to be working on a set of DICOM/WSI files from
00186         // LEICA
00187         gdcWarningMacro( "Trying to fix the even-but-odd value length bug #1" );
00188         gdcassert( Fragments.size() );
00189         const size_t lastf = Fragments.size() - 1;
00190         const ByteValue *bv = Fragments[ lastf ].GetByteValue();

```

```

00190     const char *a = bv->GetPointer();
00191     gdcmlAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 1 ] == 0xfe );
00192     Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
00193     is.seekg( -9, std::ios::cur );
00194     gdcmlAssert( is.good() );
00195     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00196     {
00197         gdcmlDebugMacro( "Frag: " « frag );
00198         Fragments.push_back( frag );
00199     }
00200     gdcmlAssert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00201 }
00202 // 4. LEICA/WSI (bis)
00203 else if ( frag.GetTag().GetGroup() == 0xe000 )
00204 {
00205     // Looks like there is a mess with offset and odd byte array
00206     // We are going first to backtrack one byte back, and then use a
00207     // ReadBacktrack function which in turn may backtrack up to 10 bytes
00208     // backward. This appears to be working on a set of DICOM/WSI files from
00209     // LEICA
00210     gdcmlWarningMacro( "Trying to fix the even-but-odd value length bug #2" );
00211     gdcmlAssert( Fragments.size() );
00212     const size_t lastf = Fragments.size() - 1;
00213     const ByteValue *bv = Fragments[ lastf ].GetByteValue();
00214     const char *a = bv->GetPointer();
00215     gdcmlAssertAlwaysMacro( (unsigned char)a[ bv->GetLength() - 2 ] == 0xfe );
00216     Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 2 );
00217     is.seekg( -10, std::ios::cur );
00218     gdcmlAssert( is.good() );
00219     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00220     {
00221         gdcmlDebugMacro( "Frag: " « frag );
00222         Fragments.push_back( frag );
00223     }
00224     gdcmlAssert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00225 }
00226 // 5. LEICA/WSI (ter)
00227 else if ( ( frag.GetTag().GetGroup() & 0x00ff ) == 0x00e0
00228     && ( frag.GetTag().GetElement() & 0xff00 ) == 0x0000 )
00229 {
00230     // Looks like there is a mess with offset and odd byte array
00231     // We are going first to backtrack one byte back, and then use a
00232     // ReadBacktrack function which in turn may backtrack up to 10 bytes
00233     // backward. This appears to be working on a set of DICOM/WSI files from
00234     // LEICA
00235     gdcmlWarningMacro( "Trying to fix the even-but-odd value length bug #3" );
00236     gdcmlAssert( Fragments.size() );
00237     const size_t lastf = Fragments.size() - 1;
00238     const ByteValue *bv = Fragments[ lastf ].GetByteValue();
00239     const char *a = bv->GetPointer();
00240     gdcmlAssertAlwaysMacro( bv->GetLength() >= 3 && (unsigned char)a[ bv->GetLength() - 3 ] == 0xfe );
00241     Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 3 );
00242     is.seekg( -11, std::ios::cur );
00243     gdcmlAssert( is.good() );
00244     while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
00245     {
00246         gdcmlDebugMacro( "Frag: " « frag );
00247         Fragments.push_back( frag );
00248     }
00249     gdcmlAssert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
00250 }
00251 else
00252 {
00253     // 3. gdcml-JPEG-LossLess3a.dcm: easy case, an extra tag was found
00254     // instead of terminator (eof is the next char)
00255     gdcmlWarningMacro( "Reading failed at Tag:" « frag.GetTag() « " Index #"
00256         « Fragments.size() « " Offset " « is.tell() « ". Use file at own risk."
00257         « ex.what() );
00258 }
00259 #endif /* GDCM_SUPPORT_BROKEN_IMPLEMENTATION */
00260 }
00261
00262 return is;
00263 }
00264
00265 template <typename TSwap>
00266 std::ostream const &Write(std::ostream &os) const
00267 {
00268     if( !Table.Write<TSwap>(os) )
00269     {
00270         gdcmlAssert(0 && "Should not happen");

```



```

00271     return os;
00272 }
00273 for(ConstIterator it = Begin(); it != End(); ++it)
00274 {
00275     it->Write<TSwap>(os);
00276 }
00277 // seq del item is not stored, write it !
00278 const Tag seqDelItem(0xfffe,0xe0dd);
00279 seqDelItem.Write<TSwap>(os);
00280 VL zero = 0;
00281 zero.Write<TSwap>(os);
00282
00283 return os;
00284 }
00285
00286 // #if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA)
00287 // For now leave it there, this does not make sense in the C++ layer
00288 // Create a new object
00289 static SmartPointer<SequenceOfFragments> New()
00290 {
00291     return new SequenceOfFragments();
00292 }
00293 // #endif
00294
00295 protected:
00296 public:
00297 void Print(std::ostream &os) const override {
00298     os << "SQ L= " << SequenceLengthField << "\n";
00299     os << "Table:" << Table << "\n";
00300     for(ConstIterator it = Begin(); it != End(); ++it)
00301     {
00302         os << " " << *it << "\n";
00303     }
00304     gdcm_assert( SequenceLengthField.IsUndefined() );
00305     {
00306         const Tag seqDelItem(0xfffe,0xe0dd);
00307         VL zero = 0;
00308         os << seqDelItem;
00309         os << "\t" << zero;
00310     }
00311 }
00312 bool operator==(const Value &val) const override
00313 {
00314     const SequenceOfFragments &sqf = dynamic_cast<const SequenceOfFragments&>(val);
00315     return Table == sqf.Table &&
00316         SequenceLengthField == sqf.SequenceLengthField &&
00317         Fragments == sqf.Fragments;
00318 }
00319
00320 private:
00321 BasicOffsetTable Table;
00322 VL SequenceLengthField;
00324 FragmentVector Fragments;
00325
00326 private:
00327 bool FillFragmentWithJPEG( Fragment & frag, std::istream & is );
00328 };
00329
00334
00335 } // end namespace gdcm_ns
00336
00337 #endif //GDCMSEQUENCEOFFRAGMENTS_H

```

13.173 gdcmSequenceOfItems.h File Reference

```

#include "gdcmValue.h"
#include "gdcmItem.h"
#include <vector>
#include <cstring>
#include "gdcmSequenceOfItems.txx"

```



```

00039 class GDCM_EXPORT SequenceOfItems : public Value
00040 {
00041 public:
00042 // Typdefs:
00043 typedef std::vector< Item > ItemVector;
00044 typedef ItemVector::size_type SizeType;
00045 typedef ItemVector::iterator Iterator;
00046 typedef ItemVector::const_iterator ConstIterator;
00047 Iterator Begin() { return Items.begin(); }
00048 Iterator End() { return Items.end(); }
00049 ConstIterator Begin() const { return Items.begin(); }
00050 ConstIterator End() const { return Items.end(); }
00051
00053 SequenceOfItems():SequenceLengthField(0xFFFFFFFF) { }
00054 //SequenceOfItems(VL const &vl = 0xFFFFFFFF):SequenceLengthField(vl),NType(type) { }
00055
00057 VL GetLength() const override { return SequenceLengthField; }
00059 void SetLength(VL length) override {
00060     SequenceLengthField = length;
00061 }
00063 void SetLengthToUndefined();
00065 bool IsUndefinedLength() const {
00066     return SequenceLengthField.IsUndefined();
00067 }
00068
00069 template <typename TDE>
00070 VL ComputeLength() const;
00071
00073 void Clear() override;
00074
00076 void AddItem(Item const &item);
00077
00079 Item & AddNewUndefinedLengthItem();
00080
00083 bool RemoveItemByIndex( const SizeType index );
00084
00085 bool IsEmpty() const { return Items.empty(); }
00086 SizeType GetNumberOfItems() const { return Items.size(); }
00087 void SetNumberOfItems(SizeType n) { Items.resize(n); }
00088
00089 /* WARNING: first item is #1 (see DICOM standard)
00090 * Each Item shall be implicitly assigned an ordinal position starting with the value 1 for the
00091 * first Item in the Sequence, and incremented by 1 with each subsequent Item. The last Item in the
00092 * Sequence shall have an ordinal position equal to the number of Items in the Sequence.
00093 */
00094 const Item &GetItem(SizeType position) const;
00095 Item &GetItem(SizeType position);
00096
00097 SequenceOfItems &operator=(const SequenceOfItems &val) {
00098     SequenceLengthField = val.SequenceLengthField;
00099     Items = val.Items;
00100     return *this;
00101 }
00102
00103 template <typename TDE, typename TSwap>
00104 std::istream &Read(std::istream &is, bool readvalues = true)
00105 {
00106     (void)readvalues;
00107     const Tag seqDelItem(0xfffe,0xe0dd);
00108     if( SequenceLengthField.IsUndefined() )
00109     {
00110         Item item;
00111         while( item.Read<TDE,TSwap>(is) && item.GetTag() != seqDelItem )
00112         {
00113             //gdcmDebugMacro( "Item: " « item );
00114             gdcm_assert( item.GetTag() != seqDelItem );
00115             Items.push_back( item );
00116             item.Clear();
00117         }
00118         //gdcm_assert( item.GetTag() == seqDelItem && item.GetVL() == 0 );
00119     }
00120     else
00121     {
00122         Item item;
00123         VL l = 0;
00124         //is.seekg( SequenceLengthField, std::ios::cur ); return is;
00125         while( l != SequenceLengthField )
00126         {
00127             try
00128             {
00129                 item.Read<TDE,TSwap>(is);

```

```

00130     }
00131     catch( Exception &ex )
00132     {
00133         if( strcmp( ex.GetDescription(), "Changed Length" ) == 0 )
00134         {
00135             VL newlength = 1 + item.template GetLength<TDE>();
00136             if( newlength > SequenceLengthField )
00137             {
00138                 // BogusItemAndSequenceLength.dcm
00139                 gdcmWarningMacro( "SQ length is wrong" );
00140                 SequenceLengthField = newlength;
00141             }
00142         }
00143         else
00144         {
00145             throw ex;
00146         }
00147     }
00148 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00149     if( item.GetTag() == seqDelItem )
00150     {
00151         gdcmWarningMacro( "SeqDelItem found in defined length Sequence. Skipping" );
00152         gdcm_assert( item.GetVL() == 0 );
00153         gdcm_assert( item.GetNestedDataSet().Size() == 0 );
00154         // we need to pay attention that the length of the Sequence of Items will be wrong
00155         // this way. Indeed by not adding this item we are changing the size of this sq
00156     }
00157     else // Not a seq del item marker
00158 #endif
00159     {
00160         // By design we never load them. If we were to load those attribute
00161         // as normal item it would become very complex to convert a sequence
00162         // from defined length to undefined length with the risk to write two
00163         // seq del marker
00164         Items.push_back( item );
00165     }
00166     l += item.template GetLength<TDE>();
00167     if( l > SequenceLengthField )
00168     {
00169         gdcmDebugMacro( "Found: Length of Item larger than expected" );
00170         throw "Length of Item larger than expected";
00171     }
00172     gdcm_assert( l <= SequenceLengthField );
00173     //std::cerr << "sqi debug len: " << is.tell() << " " << l << " " << SequenceLengthField << std::endl;
00174 #ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
00175     // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
00176     // (0x2005, 0x1080): for some reason computation of length fails...
00177     if( SequenceLengthField == 778 && l == 774 )
00178     {
00179         gdcmWarningMacro( "PMS: Super bad hack" );
00180         SequenceLengthField = l;
00181         throw Exception( "Wrong Length" );
00182         //l = SequenceLengthField;
00183     }
00184     // Bug_Philips_ItemTag_3F3F
00185     // (0x2005, 0x1080): Because we do not handle fully the bug at the item
00186     // level we need to check here too
00187     else if ( SequenceLengthField == 444 && l == 3*71 )
00188     {
00189         // This one is a double bug. Item length is wrong and impact SQ length
00190         gdcmWarningMacro( "PMS: Super bad hack" );
00191         l = SequenceLengthField;
00192     }
00193 #endif
00194     }
00195     gdcm_assert( l == SequenceLengthField );
00196 }
00197 return is;
00198 }
00199
00200 template <typename TDE,typename TSwap>
00201 std::ostream const &Write( std::ostream &os ) const
00202 {
00203     typename ItemVector::const_iterator it = Items.begin();
00204     for( ; it != Items.end(); ++it )
00205     {
00206         it->Write<TDE,TSwap>(os);
00207     }
00208     if( SequenceLengthField.IsUndefined() )
00209     {
00210         // seq del item is not stored, write it !

```

```

00211     const Tag seqDelItem(0xfffe,0xe0dd);
00212     seqDelItem.Write<TSwap>(os);
00213     VL zero = 0;
00214     zero.Write<TSwap>(os);
00215 }
00216
00217 return os;
00218 }
00219
00220 //protected:
00221 void Print(std::ostream &os) const override {
00222     os << "\\t(" << SequenceLengthField << ")\n";
00223     ItemVector::const_iterator it =
00224         Items.begin();
00225     for(;it != Items.end(); ++it)
00226     {
00227         os << " " << *it;
00228     }
00229     if( SequenceLengthField.IsUndefined() )
00230     {
00231         const Tag seqDelItem(0xfffe,0xe0dd);
00232         VL zero = 0;
00233         os << seqDelItem;
00234         os << "\\t" << zero;
00235     }
00236 }
00237
00238 static SmartPointer<SequenceOfItems> New()
00239 {
00240     return new SequenceOfItems;
00241 }
00242 bool FindDataElement(const Tag &t) const;
00243
00244 bool operator==(const Value &val) const override
00245 {
00246     const SequenceOfItems &sqi = dynamic_cast<const SequenceOfItems&>(val);
00247     return SequenceLengthField == sqi.SequenceLengthField &&
00248         Items == sqi.Items;
00249 }
00250
00251 private:
00252 public:
00253     VL SequenceLengthField;
00254     ItemVector Items;
00255 };
00256
00257 } // end namespace gdcm_ns
00258
00259 #include "gdcmSequenceOfItems.txx"
00260
00261 #endif //GDCMSEQUENCEOFITEMS_H

```

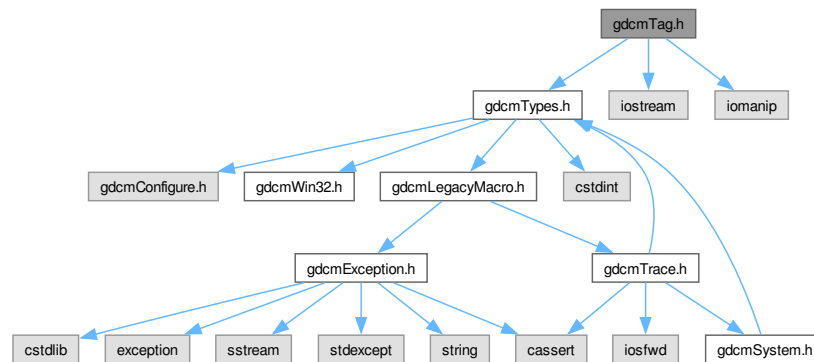
13.175 gdcmTag.h File Reference

```

#include "gdcmTypes.h"
#include <iostream>
#include <iomanip>

```

Include dependency graph for `gdcmTag.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Tag`
Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)`
- `std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)`

13.176 gdcmtag.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmtag.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTAG_H
00015 #define GDCMTAG_H
00016
00017 #include "gdcmtypes.h"
00018
00019 #include <iostream>
00020 #include <iomanip>
00021
00022 namespace gdcmtag
00023 {
00024
00025 class GDCM_EXPORT Tag
00026 {
00027 public:
00028     Tag(uint16_t group, uint16_t element) {
00029         ElementTag.tags[0] = group; ElementTag.tags[1] = element;
00030     }
00031     Tag(uint32_t tag = 0) {
00032         SetElementTag(tag);
00033     }
00034
00035 friend std::ostream& operator<<(std::ostream &_os, const Tag &_val);
00036 friend std::istream& operator>>(std::istream &_is, Tag &_val);
00037
00038 uint16_t GetGroup() const { return ElementTag.tags[0]; }
00039 uint16_t GetElement() const { return ElementTag.tags[1]; }
00040 void SetGroup(uint16_t group) { ElementTag.tags[0] = group; }
00041 void SetElement(uint16_t element) { ElementTag.tags[1] = element; }
00042 void SetElementTag(uint16_t group, uint16_t element) {
00043     ElementTag.tags[0] = group; ElementTag.tags[1] = element;
00044 }
00045
00046 uint32_t GetElementTag() const {
00047 #ifndef GDCM_WORDS_BIGENDIAN
00048     return (ElementTag.tag<<16) | (ElementTag.tag>>16);
00049 #else
00050     return ElementTag.tag;
00051 #endif
00052 }
00053
00054 void SetElementTag(uint32_t tag) {
00055 #ifndef GDCM_WORDS_BIGENDIAN
00056     tag = ( (tag<<16) | (tag>>16) );
00057 #endif
00058     ElementTag.tag = tag;
00059 }
00060
00061 const uint16_t &operator[](const unsigned int &_id) const
00062 {
00063     gdcmtag_assert(_id<2);
00064     return ElementTag.tags[_id];
00065 }
00066
00067 uint16_t &operator[](const unsigned int &_id)
00068 {
00069     gdcmtag_assert(_id<2);
00070     return ElementTag.tags[_id];
00071 }
00072
00073 Tag &operator=(const Tag &_val)
00074 {

```

```

00099     ElementTag.tag = _val.ElementTag.tag;
00100     return *this;
00101 }
00102
00103 bool operator==(const Tag &_val) const
00104 {
00105     return ElementTag.tag == _val.ElementTag.tag;
00106 }
00107 bool operator!=(const Tag &_val) const
00108 {
00109     return ElementTag.tag != _val.ElementTag.tag;
00110 }
00111
00112 // FIXME FIXME FIXME TODO
00113 // the following is pretty dumb. Since we have control over who is group
00114 // and who is element, we should reverse them in little endian and big endian case
00115 // since what we really want is fast comparison and not guarantee that group is in #0
00116 // ...
00117 bool operator<(const Tag &_val) const
00118 {
00119     #ifndef GDCM_WORDS_BIGENDIAN
00120     if( ElementTag.tags[0] < _val.ElementTag.tags[0] )
00121         return true;
00122     if( ElementTag.tags[0] == _val.ElementTag.tags[0]
00123         && ElementTag.tags[1] < _val.ElementTag.tags[1] )
00124         return true;
00125     return false;
00126     #else
00127     // Plain comparison is enough!
00128     return ( ElementTag.tag < _val.ElementTag.tag );
00129     #endif
00130 }
00131 bool operator<=(const Tag &t2) const
00132 {
00133     const Tag &t1 = *this;
00134     return t1 == t2 || t1 < t2;
00135 }
00136
00137 Tag(const Tag &_val)
00138 {
00139     ElementTag.tag = _val.ElementTag.tag;
00140 }
00141
00142 uint32_t GetLength() const { return 4; }
00143
00144 bool IsPublic() const { return !(ElementTag.tags[0] % 2); }
00145
00146 bool IsPrivate() const { return !IsPublic(); }
00147
00148 //-----
00149 template <typename TSwap>
00150 std::istream &Read(std::istream &is)
00151 {
00152     if( is.read(ElementTag.bytes, 4) )
00153         TSwap::SwapArray(ElementTag.tags, 2);
00154     return is;
00155 }
00156
00157 template <typename TSwap>
00158 const std::ostream &Write(std::ostream &os) const
00159 {
00160     uint16_t copy[2];
00161     copy[0] = ElementTag.tags[0];
00162     copy[1] = ElementTag.tags[1];
00163     TSwap::SwapArray(copy, 2);
00164     return os.write((char*)&copy, 4);
00165 }
00166
00167 Tag GetPrivateCreator() const
00168 {
00169     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
00170     // eg: 0x0123,0x1425 -> 0x0123,0x0014
00171     if( IsPrivate() && !IsPrivateCreator() )
00172     {
00173         Tag r = *this;
00174         r.SetElement( (uint16_t)(GetElement() » 8) );
00175         return r;
00176     }
00177     if( IsPrivateCreator() ) return *this;
00178     return Tag(0x0,0x0);
00179 }
00180
00181 void SetPrivateCreator(Tag const &t)

```



```

00194 {
00195     // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
00196     // eg: 0x0123,0x0045 -> 0x0123,0x4567
00197     gdcmt_assert( t.IsPrivate() /*&& t.IsPrivateCreator()*/ );
00198     const uint16_t element = (uint16_t)(t.GetElement() « 8);
00199     const uint16_t base = (uint16_t)(GetElement() « 8);
00200     SetElement( (uint16_t)((base » 8) + element) );
00201     SetGroup( t.GetGroup() );
00202 }
00203
00206 bool IsPrivateCreator() const
00207 {
00208     return IsPrivate() && (GetElement() <= 0xFF && GetElement() >= 0x10);
00209 }
00210
00212 bool IsIllegal() const
00213 {
00214     // DICOM reserved those groups:
00215     return GetGroup() == 0x0001 || GetGroup() == 0x0003 || GetGroup() == 0x0005 || GetGroup() == 0x0007
00216     // This is a very special case, in private group, one cannot use element [0x01,0x09] ...
00217     // || (IsPrivate() && !IsPrivateCreator() && !IsGroupLength());
00218     // || (IsPrivate() && GetElement() > 0x0 && GetElement() < 0x10 );
00219 }
00220
00222 bool IsGroupLength() const
00223 {
00224     return GetElement() == 0x0;
00225 }
00226
00228 bool IsGroupXX(const Tag &t) const
00229 {
00230     if( t.GetElement() == GetElement() )
00231     {
00232         if( t.IsPrivate() ) return false;
00233         uint16_t group = (uint16_t)((GetGroup() » 8) « 8);
00234         return group == t.GetGroup();
00235     }
00236     return false;
00237 }
00238
00244 bool ReadFromCommaSeparatedString(const char *str);
00245
00248 bool ReadFromContinuousString(const char *str);
00249
00252 std::string PrintAsContinuousString() const;
00253
00255 std::string PrintAsContinuousUpperCaseString() const;
00256
00259 bool ReadFromPipeSeparatedString(const char *str);
00260
00263 std::string PrintAsPipeSeparatedString() const;
00264
00265 private:
00266     union { uint32_t tag; uint16_t tags[2]; char bytes[4]; } ElementTag;
00267 };
00268 //-----
00269 inline std::istream& operator>(std::istream &_is, Tag &_val)
00270 {
00271     char c;
00272     _is » c;
00273     uint16_t a, b;
00274     _is » std::hex » a;
00275     // _is » std::hex » _val[0];
00276     // _is » std::hex » _val.ElementTag.tags[0];
00277     _is » c;
00278     // _is » _val[1];
00279     // _is » std::hex » _val.ElementTag.tags[1];
00280     _is » std::hex » b;
00281     _is » c;
00282     _val.SetGroup( a );
00283     _val.SetElement( b );
00284     return _is;
00285 }
00286
00287 inline std::ostream& operator<(std::ostream &_os, const Tag &_val)
00288 {
00289     _os.setf( std::ios::right);
00290     _os « std::hex « '(' « std::setw( 4 ) « std::setfill( '0' )
00291     « _val[0] « ',' « std::setw( 4 ) « std::setfill( '0' )
00292     « _val[1] « ')' « std::setfill( ' ' ) « std::dec;
00293     return _os;

```

```

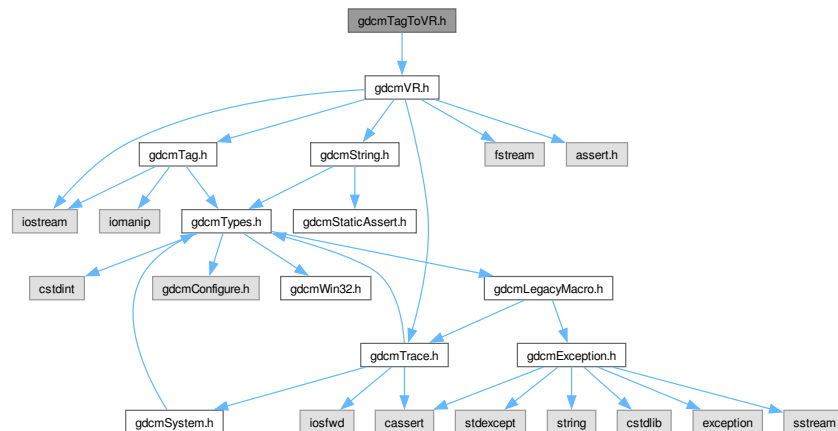
00294 }
00295
00296 } // end namespace gdcm
00297
00298 #endif //GDCMTAG_H

```

13.177 gdcmTagToVR.h File Reference

#include "gdcmVR.h"

Include dependency graph for gdcmTagToVR.h:



Namespaces

- namespace [gdcm](#)

Functions

- [VR::VRType gdcm::GetVRFromTag \(Tag const &tag\)](#)

13.178 gdcmTagToVR.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.

```

```

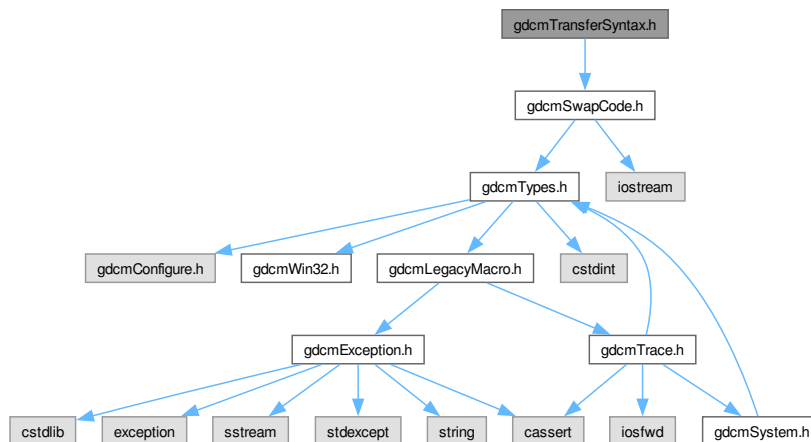
00012
00013
00014 =====*/
00014 #ifndef GDCMTAGTOVR_H
00015 #define GDCMTAGTOVR_H
00016
00017 #include "gdcmVR.h"
00018
00019 namespace gdcm
00020 {
00021     class Tag;
00022     VR::VRType GetVRFromTag( Tag const & tag );
00023 }
00024
00025 #endif // GDCMTAGTOVR_H

```

13.179 gdcmTransferSyntax.h File Reference

#include "gdcmSwapCode.h"

Include dependency graph for gdcmTransferSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::TransferSyntax`
Class to manipulate Transfer Syntax.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

13.180 gdcmmTransferSyntax.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTRANSFERSYNTAX_H
00015 #define GDCMTRANSFERSYNTAX_H
00016
00017 #include "gdcmmSwapCode.h"
00018
00019 namespace gdcmm
00020 {
00021
00039 class GDCM_EXPORT TransferSyntax
00040 {
00041 public:
00042     typedef enum {
00043         Unknown = 0,
00044         Explicit,
00045         Implicit
00046     } NegotiatedType;
00047
00048     #if 0
00049     //NOT FLEXIBLE, since forces user to update lib every time new module
00050     //comes out...
00051     // TODO
00052     typedef enum {
00053         NoSpacing = 0,
00054         PixelSpacing,
00055         ImagerPixelSpacing,
00056         PixelAspectRatio
00057     } ImageSpacingType;
00058     ImageSpacingType GetImageSpacing();
00059     #endif
00060
00061     typedef enum {
00062         ImplicitVRLittleEndian = 0,
00063         ImplicitVRBigEndianPrivateGE,
00064         ExplicitVRLittleEndian,
00065         DeflatedExplicitVRLittleEndian,
00066         ExplicitVRBigEndian,
00067         JPEGBaselineProcess1,
00068         JPEGExtendedProcess2_4,
00069         JPEGExtendedProcess3_5,
00070         JPEGSpectralSelectionProcess6_8,
00071         JPEGFullProgressionProcess10_12,
00072         JPEGLosslessProcess14,
00073         JPEGLosslessProcess14_1,
00074         JPEGLSLossless,
00075         JPEGLSNearLossless,
00076         JPEG2000Lossless,
00077         JPEG2000,
00078         JPEG2000Part2Lossless,
00079         JPEG2000Part2,
00080         RLELossless,
00081         MPEG2MainProfile,
00082         ImplicitVRBigEndianACRNEMA,

```

```

00083     WeirdPapryus,
00084     CT_private_ELE,
00085     JPIPReferenced,
00086     MPEG2MainProfileHighLevel,
00087     MPEG4AVCH264HighProfileLevel4_1,
00088     MPEG4AVCH264BDcompatibleHighProfileLevel4_1,
00089     HTJ2KLossless,
00090     HTJ2KRPCLossless,
00091     HTJ2K,
00092     DeflatedImageFrameCompression,
00093     TS_END
00094 } TSType;
00095
00096 // Return the string as written in the official DICOM dict from
00097 // a custom enum type
00098 static const char* GetTSString(TSType ts);
00099 static TSType GetTSType(const char *str);
00100
00101 NegotiatedType GetNegociatedType() const;
00102
00106 SwapCode GetSwapCode() const;
00107
00108 bool IsValid() const { return TSField != TS_END; }
00109
00110 operator TSType () const { return TSField; }
00111
00112 // FIXME: ImplicitVRLittleEndian used to be the default, but nowadays
00113 // this is rather the ExplicitVRLittleEndian instead...should be change the default ?
00114 TransferSyntax(TSType type = ImplicitVRLittleEndian):TSField(type) {}
00115
00116 // return if dataset is encoded or not (Deflate Explicit VR)
00117 bool IsEncoded() const;
00118
00119 bool IsImplicit() const;
00120 bool IsExplicit() const;
00121
00122 bool IsEncapsulated() const;
00123
00125 bool IsLossy() const;
00127 bool IsLossless() const;
00129 bool CanStoreLossy() const;
00130
00131 const char *GetString() const { return TransferSyntax::GetTSString(TSField); }
00132
00133 friend std::ostream &operator<<(std::ostream &os, const TransferSyntax &ts);
00134 private:
00135 // DO NOT EXPOSE the following. Internal details of TransferSyntax
00136 bool IsImplicit(TSType ts) const;
00137 bool IsExplicit(TSType ts) const;
00138 bool IsLittleEndian(TSType ts) const;
00139 bool IsBigEndian(TSType ts) const;
00140
00141 TSType TSField;
00142 };
00143 //-----
00144 inline std::ostream &operator<<(std::ostream &_os, const TransferSyntax &ts)
00145 {
00146     _os << TransferSyntax::GetTSString(ts);
00147     return _os;
00148 }
00149 }
00150
00151 } // end namespace gdcm
00152
00153 #endif //GDCMTRANSFERSYNTAX_H

```

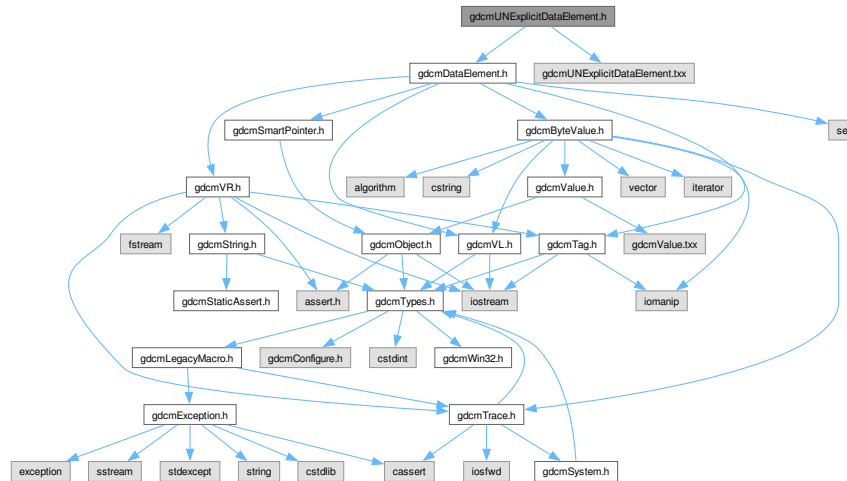
13.181 gdcmUNExplicitDataElement.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"

```

Include dependency graph for `gdcmUNExplicitDataElement.h`:



Classes

- class `gdcm::UNExplicitDataElement`
Class to read/write a `DataElement` as UNExplicit Data `Element`.

Namespaces

- namespace `gdcm`

13.182 gdcmUNExplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMUNEXPLICITDATAELEMENT_H
00015  #define GDCMUNEXPLICITDATAELEMENT_H
00016
00017  #include "gdcmDataElement.h"
00018
00019  namespace gdcm
00020  {
00021  // Data Element (UNExplicit)

```

```

00026 class GDCM_EXPORT UNExplicitDataElement : public DataElement
00027 {
00028 public:
00029     VL GetLength() const;
00030
00031     template <typename TSwap>
00032     std::istream &Read(std::istream &is);
00033
00034     template <typename TSwap>
00035     std::istream &ReadPreValue(std::istream &is);
00036
00037     template <typename TSwap>
00038     std::istream &ReadValue(std::istream &is, bool readvalues = true);
00039
00040     template <typename TSwap>
00041     std::istream &ReadWithLength(std::istream &is, VL & length);
00042
00043     // PURPOSELY do not provide an implementation for writing !
00044     //template <typename TSwap>
00045     //const std::ostream &Write(std::ostream &os) const;
00046 };
00047
00048 } // end namespace gdcm
00049
00050 #include "gdcmUNExplicitDataElement.txx"
00051
00052 #endif //GDCMUNEXPLICITDATAELEMENT_H

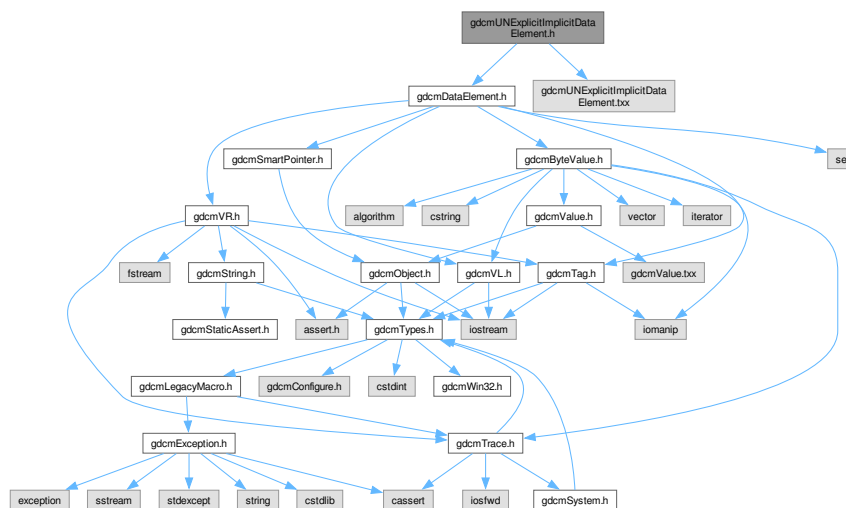
```

13.183 gdcmUNExplicitImplicitDataElement.h File Reference

#include "gdcmDataElement.h"

#include "gdcmUNExplicitImplicitDataElement.txx"

Include dependency graph for gdcmUNExplicitImplicitDataElement.h:



Classes

- class [gdcm::UNExplicitImplicitDataElement](#)

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

Namespaces

- namespace [gdcm](#)

13.184 gdcmUNExplicitImplicitDataElement.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMUNEXPLICITIMPLICITDATAELEMENT_H
00015  #define GDCMUNEXPLICITIMPLICITDATAELEMENT_H
00016  00017  #include "gdcmDataElement.h"
00018  00019  namespace gdcm
00020  {
00021  // Data Element (ExplicitImplicit)
00022  class GDCM_EXPORT UNExplicitImplicitDataElement : public DataElement
00023  {
00024  public:
00025  VL GetLength() const;
00026  00027  template <typename TSwap>
00028  std::istream &Read(std::istream &is);
00029  00030  template <typename TSwap>
00031  std::istream &ReadPreValue(std::istream &is);
00032  00033  template <typename TSwap>
00034  std::istream &ReadValue(std::istream &is);
00035  00036  // PURPOSELY do not provide an implementation for writing !
00037  //template <typename TSwap>
00038  //const std::ostream &Write(std::ostream &os) const;
00039  };
00040  00041  } // end namespace gdcm
00042  00043  #include "gdcmUNExplicitImplicitDataElement.txx"
00044  00045  #endif //GDCMUNEXPLICITIMPLICITDATAELEMENT_H

```

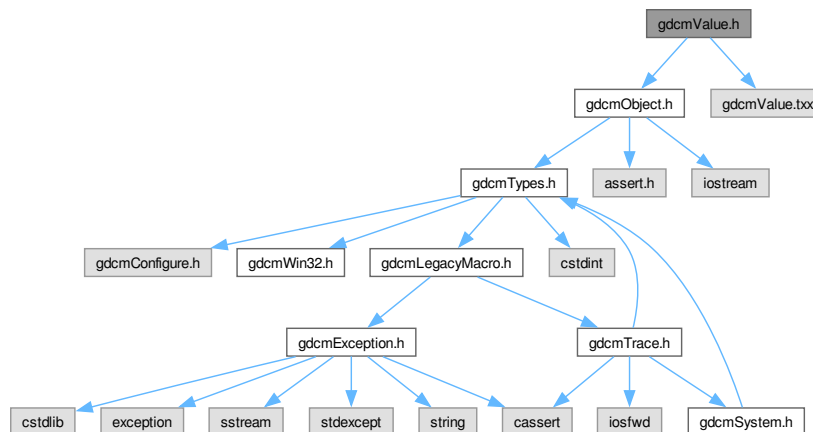
13.185 gdcmValue.h File Reference

```

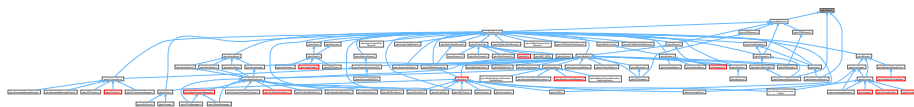
#include "gdcmObject.h"
#include "gdcmValue.txx"

```


Include dependency graph for gdcValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Value](#)
Class to represent the value of a Data [Element](#).

Namespaces

- namespace [gdcm](#)

13.186 gdcValue.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMVALUE_H
00015  #define GDCMVALUE_H
00016
00017  #include "gdcmObject.h"
00018
00019  namespace gdcm { class VL; }
00020  namespace gdcm_ns
00021  {
00022  #if !defined(SWIGPYTHON) && !defined(SWIGSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023  using namespace gdcm;
00024  #endif
00031  class GDCM_EXPORT Value : public Object
00032  {
00033  public:
00034  Value() = default;
00035  ~Value() override = default;
00036
00037  virtual VL GetLength() const = 0;
00038  virtual void SetLength(VL l) = 0;
00039
00040  virtual void Clear() = 0;
00041
00042  virtual bool operator==(const Value &val) const = 0;
00043
00044  protected:
00045  friend class DataElement;
00046  virtual void SetLengthOnly(VL l);
00047  };
00048
00049  } // end namespace gdcm_ns
00050
00051  #include "gdcmValue.txx"
00052
00053  #endif //GDCMVALUE_H

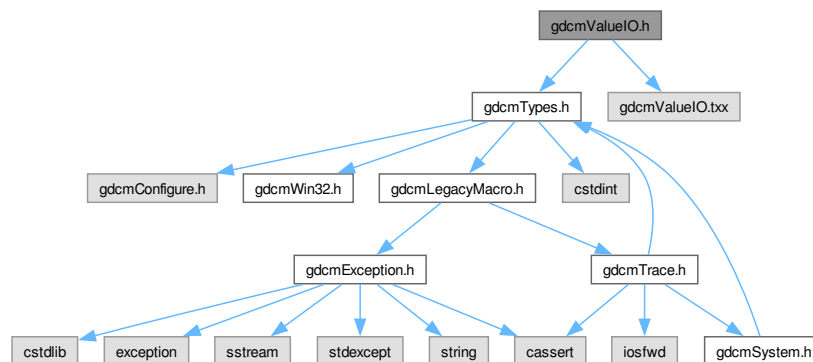
```

13.187 gdcmValueIO.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmValueIO.txx"
```

Include dependency graph for gdcmValueIO.h:



Classes

- class `gdcm::ValueIO< TDE, TSwap, TType >`

Class to dispatch template calls.

Namespaces

- namespace [gdcm](#)

13.188 gdcmValueIO.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMVALUEIO_H
00015 #define GDCMVALUEIO_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm_ns
00020 {
00021     template <typename TDE, typename TSwap, typename TType=uint8_t>
00022     class /*GDCM_EXPORT*/ ValueIO
00023     {
00024     public:
00025         static std::istream &Read(std::istream &is, Value& v, bool readvalues);
00026         static const std::ostream &Write(std::ostream &os, const Value& v);
00027     };
00028 } // end namespace gdcm_ns
00029
00030 #include "gdcmValueIO.txx"
00031 #endif //GDCMVALUEIO_H

```

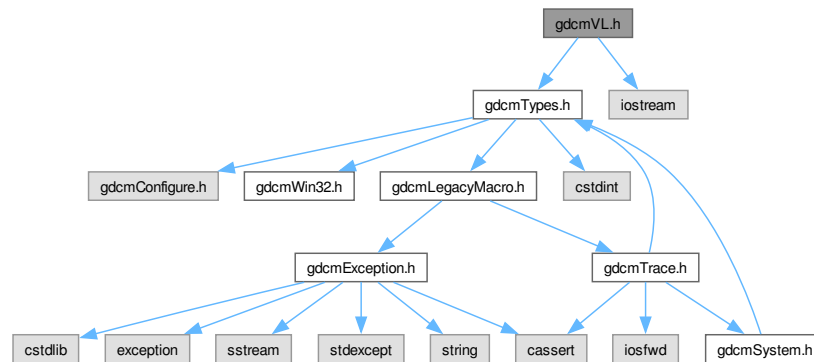
13.189 gdcmVL.h File Reference

```

#include "gdcmTypes.h"
#include <iostream>

```

Include dependency graph for `gdcmVL.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::VL`
Value Length.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)`

13.190 gdcmVL.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
  
```

```

00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMVL_H
00015 #define GDCMVL_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023
00024 class GDCM_EXPORT VL
00025 {
00026 public:
00027     typedef uint32_t Type;
00028     VL(uint32_t vl = 0) : ValueLength(vl) { }
00029
00030     // FIXME: ugly
00031     static uint32_t GetVL32Max() { return 0xFFFFFFFF; }
00032     static uint16_t GetVL16Max() { return 0xFFFF; }
00033
00034     bool IsUndefined() const {
00035         return ValueLength == 0xFFFFFFFF;
00036     }
00037     void SetToUndefined() {
00038         ValueLength = 0xFFFFFFFF;
00039     }
00040
00041     bool IsOdd() const {
00042         return !IsUndefined() && ValueLength % 2;
00043     }
00044
00045     VL& operator+=(VL const &vl) {
00046         ValueLength += vl.ValueLength;
00047         return *this;
00048     }
00049     VL& operator++() {
00050         ++ValueLength;
00051         return *this;
00052     }
00053     VL operator++(int) {
00054         uint32_t tmp(ValueLength);
00055         ++ValueLength;
00056         return tmp;
00057     }
00058
00059     operator uint32_t () const { return ValueLength; }
00060
00061     VL GetLength() const {
00062         // VL cannot know it's length...well in implicit yes...
00063         // TODO: need to check we cannot call this function from an Explicit element
00064         return 4;
00065     }
00066
00067     friend std::ostream& operator<<(std::ostream& os, const VL& vl);
00068
00069     // PURPOSELY not implemented (could not differentiate 16bits vs 32bits VL)
00070     //friend std::istream& operator>>(std::istream& is, VL& n);
00071
00072     template <typename TSwap>
00073     std::istream &Read(std::istream &is)
00074     {
00075         is.read((char*)&ValueLength, sizeof(uint32_t));
00076         TSwap::SwapArray(&ValueLength,1);
00077         return is;
00078     }
00079
00080     template <typename TSwap>
00081     std::istream &Read16(std::istream &is)
00082     {
00083         uint16_t copy;
00084         is.read((char*)&copy, sizeof(uint16_t));
00085         TSwap::SwapArray(&copy,1);
00086     }
00087
00088
00089
00090
00091
00092

```

```

00093     ValueLength = copy;
00094     gdcM_assert( ValueLength <=  65535 /*UINT16_MAX*/ ); // ?? doh !
00095     return is;
00096 }
00097
00098 template <typename TSwap>
00099 const std::ostream &Write(std::ostream &os) const
00100 {
00101     uint32_t copy = ValueLength;
00102     if( IsOdd() )
00103     {
00104         ++copy;
00105     }
00106     TSwap::SwapArray(&copy,1);
00107     return os.write((char*)&copy, sizeof(uint32_t));
00108 }
00109
00110 template <typename TSwap>
00111 const std::ostream &Write16(std::ostream &os) const
00112 {
00113     gdcM_assert( ValueLength <=  65535 /*UINT16_MAX*/ );
00114     uint16_t copy = (uint16_t)ValueLength;
00115     if( IsOdd() )
00116     {
00117         ++copy;
00118     }
00119     TSwap::SwapArray(&copy,1);
00120     return os.write((char*)&copy, sizeof(uint16_t));
00121 }
00122
00123 private:
00124     uint32_t ValueLength;
00125 };
00126 //-----
00127 inline std::ostream& operator<<(std::ostream& os, const VL& val)
00128 {
00129     os << /*std::hex <<*/ val.ValueLength;
00130     return os;
00131 }
00132
00133 } // end namespace gdcM
00134
00135 #endif //GDCMVL_H

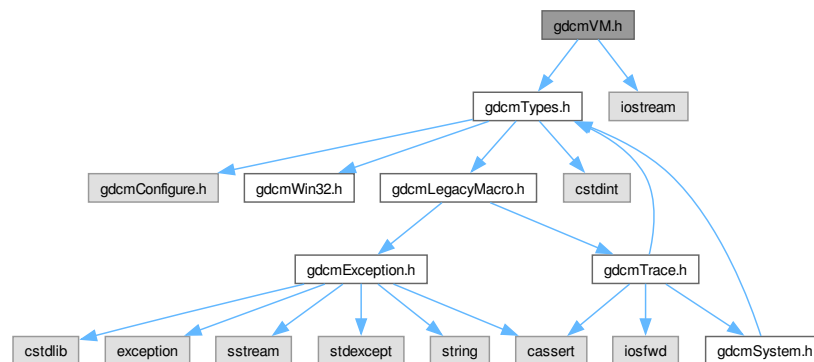
```

13.191 gdcMVM.h File Reference

#include "gdcMTypes.h"

#include <iostream>

Include dependency graph for gdcMVM.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcmm::VM](#)
[Value](#) Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Namespaces

- namespace [gdcmm](#)

Macros

- `#define` [TYPETOLENGTH](#)(type, length)

Functions

- `std::ostream &` [gdcmm::operator<<](#) (std::ostream &_os, const [VM](#) &_val)

13.191.1 Macro Definition Documentation

13.191.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(  
    type,  
    length)
```

Value:

```
template<> struct VMToLength<VM::type> \  
{ enum { Length = length }; };
```

13.192 gdcmmVM.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMMVM_H
00015 #define GDCMMVM_H
00016
00017 #include "gdcmmTypes.h"
00018 #include <iostream>
00019
00020 namespace gdcmm
00021 {
00022
00067 class GDCMM_EXPORT VM
00068 {
00069 public:
00070     typedef enum {
00071         VM0 = 0, // aka the invalid VM
00072         VM1 = 1,
00073         VM2 = 2,
00074         VM3 = 4,
00075         VM4 = 8,
00076         VM5 = 16,
00077         VM6 = 32,
00078         VM8 = 64,
00079         VM9 = 128,
00080         VM10 = 256,
00081         VM12 = 512, //1024,
00082         VM16 = 1024, //2048,
00083         VM18 = 2048, //4096,
00084         VM24 = 4096, //8192,
00085         VM28 = 8192, //16384,
00086         VM32 = 16384, //32768,
00087         VM35 = 32768, //65536,
00088         VM99 = 65536, //131072,
00089         VM256 = 131072, //262144,
00090         VM1_2 = VM1 | VM2,
00091         VM1_3 = VM1 | VM2 | VM3,
00092         VM1_4 = VM1 | VM2 | VM3 | VM4,
00093         VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5,
00094         VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8,
00095         // The following need some work:
00096         VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32,
00097         VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99,
00098         VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00099         VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256,
00100         VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00101         VM3_4 = VM3 | VM4,
00102         VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256,
00103         VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00104         VM4_4n = VM4 | VM16 | VM24 | VM32 | VM256,
00105         VM6_6n = VM6 | VM12 | VM18 | VM24,
00106         VM6_n = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
00107         VM7_7n,
00108         VM30_30n,
00109         VM47_47n,
00110         VM_END = VM1_n + 1 // Custom tag to count number of entry
00111     } VMType;
00112
00115     static const char* GetVMString(VMType vm);
00116     static VMType GetVMType(const char *vm);
00117
00120     static bool IsValid(int vm1, VMType vm2);
00121     //bool IsValid() { return VMField != VM0 && VMField < VM_END; }

```



```

00122
00126     bool Compatible(VM const &vm) const;
00127
00129     static VMType GetVMTypeFromLength(size_t length, unsigned int size);
00130     static size_t GetNumberOfElementsFromArray(const char *array, size_t length);
00131
00132     VM(VMType type = VM0):VMField(type) {}
00133     operator VMType () const { return VMField; }
00134     unsigned int GetLength() const;
00135
00136     friend std::ostream &operator<<(std::ostream &os, const VM &vm);
00137 protected:
00138     static unsigned int GetIndex(VMType vm);
00139
00140 private:
00141     VMType VMField;
00142 };
00143 //-----
00144 inline std::ostream& operator<<(std::ostream& _os, const VM &_val)
00145 {
00146     gdcmm_assert( VM::GetVMString(_val) );
00147     _os << VM::GetVMString(_val);
00148     return _os;
00149 }
00150
00151 //template <int TVM> struct LengthToVM;
00152 //template <> struct LengthToVM<1>
00153 //{ enum { TVM = VM::VM1 }; };
00154
00155 template<int T> struct VMToLength;
00156 #define TYPETOLENGTH(type,length) \
00157     template<> struct VMToLength<VM::type> \
00158     { enum { Length = length }; };
00159 // TODO: Could be generated from XML file
00160 //TYPETOLENGTH(VM0,1)
00161 TYPETOLENGTH(VM1,1)
00162 TYPETOLENGTH(VM2,2)
00163 TYPETOLENGTH(VM3,3)
00164 TYPETOLENGTH(VM4,4)
00165 TYPETOLENGTH(VM5,5)
00166 TYPETOLENGTH(VM6,6)
00167 TYPETOLENGTH(VM8,8)
00168 TYPETOLENGTH(VM9,9)
00169 TYPETOLENGTH(VM10,10)
00170 TYPETOLENGTH(VM12,12)
00171 TYPETOLENGTH(VM16,16)
00172 TYPETOLENGTH(VM18,18)
00173 TYPETOLENGTH(VM24,24)
00174 TYPETOLENGTH(VM28,28)
00175 TYPETOLENGTH(VM32,32)
00176 TYPETOLENGTH(VM35,35)
00177 TYPETOLENGTH(VM99,99)
00178 TYPETOLENGTH(VM256,256)
00179 //TYPETOLENGTH(VM1_2,2)
00180 //TYPETOLENGTH(VM1_3,3)
00181 //TYPETOLENGTH(VM1_8,8)
00182 //TYPETOLENGTH(VM1_32,32)
00183 //TYPETOLENGTH(VM1_99,99)
00184 //TYPETOLENGTH(VM1_n,
00185 //TYPETOLENGTH(VM2_2n,
00186 //TYPETOLENGTH(VM2_n,
00187 //TYPETOLENGTH(VM3_3n,
00188 //TYPETOLENGTH(VM3_n,
00189
00190 } // end namespace gdcmm
00191
00192 #endif //GDCMMVM_H

```

13.193 gdcmmVR.h File Reference

```

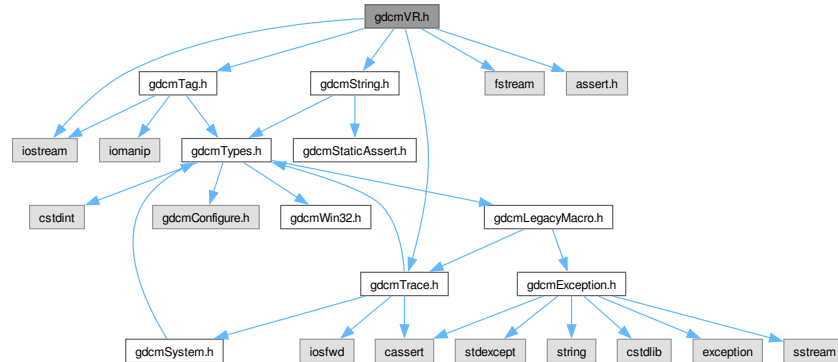
#include "gdcmmTag.h"
#include "gdcmmTrace.h"
#include "gdcmmString.h"
#include <iostream>

```

```
#include <fstream>
```

```
#include <assert.h>
```

Include dependency graph for `gdcVR.h`:



This graph shows which files directly or indirectly include this file:



Classes

- struct `gdc::UI`
- class `gdc::VR`
VR class.

Namespaces

- namespace `gdc`

Macros

- `#define` `TYPETOENCODING`(type, rep, rtype)
- `#define` `VRTypeTemplateCase`(type)

Typedefs

- typedef [String](#)<'\\', 16 > [gdcm::AECComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::ASComp](#)
- typedef [String](#)<'\\', 16 > [gdcm::CSComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::DAComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::DTComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::LOComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::LTComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::PNComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::SHComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::STComp](#)
- typedef [String](#)<'\\', 16 > [gdcm::TMComp](#)
- typedef [String](#)<'\\', 4294967294 > [gdcm::UCComp](#)
- typedef [String](#)<'\\', 64, 0 > [gdcm::UIComp](#)
- typedef [String](#)<'\\', 4294967294 > [gdcm::URComp](#)
- typedef [String](#)<'\\', 64 > [gdcm::UTComp](#)

Functions

- [std::ostream](#) & [gdcm::operator<<](#) ([std::ostream](#) &_os, const [UI](#) &_val)
- [std::ostream](#) & [gdcm::operator<<](#) ([std::ostream](#) &_os, const [VR](#) &val)

13.193.1 Macro Definition Documentation

13.193.1.1 TYPETOENCODING

```
#define TYPETOENCODING(
    type,
    rep,
    rtype)
```

Value:

```
template<> struct VRToEncoding<VR::type> \
{ enum:long long { Mode = VR::rep }; }; \
template<> struct VRToType<VR::type> \
{ typedef rtype Type; };
```

13.193.1.2 VRTypeTemplateCase

```
#define VRTypeTemplateCase(
    type)
```

Value:

```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

Referenced by [gdcm::VR::GetSize\(\)](#).

13.194 gdcmVR.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMVR_H
00015 #define GDCMVR_H
00016
00017 #include "gdcmTag.h"
00018 #include "gdcmTrace.h"
00019 #include "gdcmString.h"
00020
00021 #include <iostream>
00022 #include <fstream>
00023 #include <assert.h>
00024
00025 //these defines are here to ensure compilation on sunos gcc
00026 #if defined (CS)
00027 # undef CS
00028 #endif
00029 #if defined (DS)
00030 # undef DS
00031 #endif
00032 #if defined (SS)
00033 # undef SS
00034 #endif
00035
00036
00037 namespace gdcm
00038 {
00039
00040 class GDCM_EXPORT VR
00041 {
00042 public:
00043     enum VRType : long long {
00044         // Warning: Do not write if ( vr & VR::INVALID ) but if ( vr == VR::INVALID )
00045         INVALID = 0, // For Item/(Seq) Item Delimitation Item
00046         AE = 1,
00047         AS = 2,
00048         AT = 4,
00049         CS = 8,
00050         DA = 16,
00051         DS = 32,
00052         DT = 64,
00053         FD = 128,
00054         FL = 256,
00055         IS = 512,
00056         LO = 1024,
00057         LT = 2048,
00058         OB = 4096,
00059         OD = 134217728, // 2^27
00060         OF = 8192,
00061         OL = 268435456, // 2^28
00062         OV = 2147483648, // 2^31
00063         OW = 16384,
00064         PN = 32768,
00065         SH = 65536,
00066         SL = 131072,
00067         SQ = 262144,
00068         SS = 524288,
00069         ST = 1048576,
00070         SV = 4294967296, // 2^32
00071         TM = 2097152,
00072         UC = 536870912, // 2^29
00073         UI = 4194304,
00074     };
00075
00076     VRType vr;
00077     VRType vr2;
00078     VRType vr3;
00079     VRType vr4;
00080     VRType vr5;
00081     VRType vr6;
00082     VRType vr7;
00083     VRType vr8;
00084     VRType vr9;
00085     VRType vr10;
00086     VRType vr11;
00087     VRType vr12;
00088     VRType vr13;
00089     VRType vr14;
00090     VRType vr15;
00091     VRType vr16;
00092     VRType vr17;
00093     VRType vr18;
00094     VRType vr19;
00095     VRType vr20;
00096     VRType vr21;
00097     VRType vr22;
00098     VRType vr23;
00099     VRType vr24;
00100     VRType vr25;
00101     VRType vr26;
00102     VRType vr27;
00103     VRType vr28;
00104     VRType vr29;
00105     VRType vr30;
00106     VRType vr31;
00107     VRType vr32;
00108     VRType vr33;
00109     VRType vr34;
00110     VRType vr35;
00111     VRType vr36;
00112     VRType vr37;
00113     VRType vr38;
00114     VRType vr39;
00115     VRType vr40;
00116     VRType vr41;
00117     VRType vr42;
00118     VRType vr43;
00119     VRType vr44;
00120     VRType vr45;
00121     VRType vr46;
00122     VRType vr47;
00123     VRType vr48;
00124     VRType vr49;
00125     VRType vr50;
00126     VRType vr51;
00127     VRType vr52;
00128     VRType vr53;
00129     VRType vr54;
00130     VRType vr55;
00131     VRType vr56;
00132     VRType vr57;
00133     VRType vr58;
00134     VRType vr59;
00135     VRType vr60;
00136     VRType vr61;
00137     VRType vr62;
00138     VRType vr63;
00139     VRType vr64;
00140     VRType vr65;
00141     VRType vr66;
00142     VRType vr67;
00143     VRType vr68;
00144     VRType vr69;
00145     VRType vr70;
00146     VRType vr71;
00147     VRType vr72;
00148     VRType vr73;
00149     VRType vr74;
00150     VRType vr75;
00151     VRType vr76;
00152     VRType vr77;
00153     VRType vr78;
00154     VRType vr79;
00155     VRType vr80;
00156     VRType vr81;
00157     VRType vr82;
00158     VRType vr83;
00159     VRType vr84;
00160     VRType vr85;
00161     VRType vr86;
00162     VRType vr87;
00163     VRType vr88;
00164     VRType vr89;
00165     VRType vr90;
00166     VRType vr91;
00167     VRType vr92;
00168     VRType vr93;
00169     VRType vr94;
00170     VRType vr95;
00171     VRType vr96;
00172     VRType vr97;
00173     VRType vr98;
00174     VRType vr99;
00175     VRType vr100;
00176     VRType vr101;
00177     VRType vr102;
00178     VRType vr103;
00179     VRType vr104;
00180     VRType vr105;
00181     VRType vr106;
00182     VRType vr107;
00183     VRType vr108;
00184     VRType vr109;
00185     VRType vr110;
00186     VRType vr111;
00187     VRType vr112;
00188     VRType vr113;
00189     VRType vr114;
00190     VRType vr115;
00191     VRType vr116;
00192     VRType vr117;
00193     VRType vr118;
00194     VRType vr119;
00195     VRType vr120;
00196     VRType vr121;
00197     VRType vr122;
00198     VRType vr123;
00199     VRType vr124;
00200     VRType vr125;
00201     VRType vr126;
00202     VRType vr127;
00203     VRType vr128;
00204     VRType vr129;
00205     VRType vr130;
00206     VRType vr131;
00207     VRType vr132;
00208     VRType vr133;
00209     VRType vr134;
00210     VRType vr135;
00211     VRType vr136;
00212     VRType vr137;
00213     VRType vr138;
00214     VRType vr139;
00215     VRType vr140;
00216     VRType vr141;
00217     VRType vr142;
00218     VRType vr143;
00219     VRType vr144;
00220     VRType vr145;
00221     VRType vr146;
00222     VRType vr147;
00223     VRType vr148;
00224     VRType vr149;
00225     VRType vr150;
00226     VRType vr151;
00227     VRType vr152;
00228     VRType vr153;
00229     VRType vr154;
00230     VRType vr155;
00231     VRType vr156;
00232     VRType vr157;
00233     VRType vr158;
00234     VRType vr159;
00235     VRType vr160;
00236     VRType vr161;
00237     VRType vr162;
00238     VRType vr163;
00239     VRType vr164;
00240     VRType vr165;
00241     VRType vr166;
00242     VRType vr167;
00243     VRType vr168;
00244     VRType vr169;
00245     VRType vr170;
00246     VRType vr171;
00247     VRType vr172;
00248     VRType vr173;
00249     VRType vr174;
00250     VRType vr175;
00251     VRType vr176;
00252     VRType vr177;
00253     VRType vr178;
00254     VRType vr179;
00255     VRType vr180;
00256     VRType vr181;
00257     VRType vr182;
00258     VRType vr183;
00259     VRType vr184;
00260     VRType vr185;
00261     VRType vr186;
00262     VRType vr187;
00263     VRType vr188;
00264     VRType vr189;
00265     VRType vr190;
00266     VRType vr191;
00267     VRType vr192;
00268     VRType vr193;
00269     VRType vr194;
00270     VRType vr195;
00271     VRType vr196;
00272     VRType vr197;
00273     VRType vr198;
00274     VRType vr199;
00275     VRType vr200;
00276     VRType vr201;
00277     VRType vr202;
00278     VRType vr203;
00279     VRType vr204;
00280     VRType vr205;
00281     VRType vr206;
00282     VRType vr207;
00283     VRType vr208;
00284     VRType vr209;
00285     VRType vr210;
00286     VRType vr211;
00287     VRType vr212;
00288     VRType vr213;
00289     VRType vr214;
00290     VRType vr215;
00291     VRType vr216;
00292     VRType vr217;
00293     VRType vr218;
00294     VRType vr219;
00295     VRType vr220;
00296     VRType vr221;
00297     VRType vr222;
00298     VRType vr223;
00299     VRType vr224;
00300     VRType vr225;
00301     VRType vr226;
00302     VRType vr227;
00303     VRType vr228;
00304     VRType vr229;
00305     VRType vr230;
00306     VRType vr231;
00307     VRType vr232;
00308     VRType vr233;
00309     VRType vr234;
00310     VRType vr235;
00311     VRType vr236;
00312     VRType vr237;
00313     VRType vr238;
00314     VRType vr239;
00315     VRType vr240;
00316     VRType vr241;
00317     VRType vr242;
00318     VRType vr243;
00319     VRType vr244;
00320     VRType vr245;
00321     VRType vr246;
00322     VRType vr247;
00323     VRType vr248;
00324     VRType vr249;
00325     VRType vr250;
00326     VRType vr251;
00327     VRType vr252;
00328     VRType vr253;
00329     VRType vr254;
00330     VRType vr255;
00331     VRType vr256;
00332     VRType vr257;
00333     VRType vr258;
00334     VRType vr259;
00335     VRType vr260;
00336     VRType vr261;
00337     VRType vr262;
00338     VRType vr263;
00339     VRType vr264;
00340     VRType vr265;
00341     VRType vr266;
00342     VRType vr267;
00343     VRType vr268;
00344     VRType vr269;
00345     VRType vr270;
00346     VRType vr271;
00347     VRType vr272;
00348     VRType vr273;
00349     VRType vr274;
00350     VRType vr275;
00351     VRType vr276;
00352     VRType vr277;
00353     VRType vr278;
00354     VRType vr279;
00355     VRType vr280;
00356     VRType vr281;
00357     VRType vr282;
00358     VRType vr283;
00359     VRType vr284;
00360     VRType vr285;
00361     VRType vr286;
00362     VRType vr287;
00363     VRType vr288;
00364     VRType vr289;
00365     VRType vr290;
00366     VRType vr291;
00367     VRType vr292;
00368     VRType vr293;
00369     VRType vr294;
00370     VRType vr295;
00371     VRType vr296;
00372     VRType vr297;
00373     VRType vr298;
00374     VRType vr299;
00375     VRType vr300;
00376     VRType vr301;
00377     VRType vr302;
00378     VRType vr303;
00379     VRType vr304;
00380     VRType vr305;
00381     VRType vr306;
00382     VRType vr307;
00383     VRType vr308;
00384     VRType vr309;
00385     VRType vr310;
00386     VRType vr311;
00387     VRType vr312;
00388     VRType vr313;
00389     VRType vr314;
00390     VRType vr315;
00391     VRType vr316;
00392     VRType vr317;
00393     VRType vr318;
00394     VRType vr319;
00395     VRType vr320;
00396     VRType vr321;
00397     VRType vr322;
00398     VRType vr323;
00399     VRType vr324;
00400     VRType vr325;
00401     VRType vr326;
00402     VRType vr327;
00403     VRType vr328;
00404     VRType vr329;
00405     VRType vr330;
00406     VRType vr331;
00407     VRType vr332;
00408     VRType vr333;
00409     VRType vr334;
00410     VRType vr335;
00411     VRType vr336;
00412     VRType vr337;
00413     VRType vr338;
00414     VRType vr339;
00415     VRType vr340;
00416     VRType vr341;
00417     VRType vr342;
00418     VRType vr343;
00419     VRType vr344;
00420     VRType vr345;
00421     VRType vr346;
00422     VRType vr347;
00423     VRType vr348;
00424     VRType vr349;
00425     VRType vr350;
00426     VRType vr351;
00427     VRType vr352;
00428     VRType vr353;
00429     VRType vr354;
00430     VRType vr355;
00431     VRType vr356;
00432     VRType vr357;
00433     VRType vr358;
00434     VRType vr359;
00435     VRType vr360;
00436     VRType vr361;
00437     VRType vr362;
00438     VRType vr363;
00439     VRType vr364;
00440     VRType vr365;
00441     VRType vr366;
00442     VRType vr367;
00443     VRType vr368;
00444     VRType vr369;
00445     VRType vr370;
00446     VRType vr371;
00447     VRType vr372;
00448     VRType vr373;
00449     VRType vr374;
00450     VRType vr375;
00451     VRType vr376;
00452     VRType vr377;
00453     VRType vr378;
00454     VRType vr379;
00455     VRType vr380;
00456     VRType vr381;
00457     VRType vr382;
00458     VRType vr383;
00459     VRType vr384;
00460     VRType vr385;
00461     VRType vr386;
00462     VRType vr387;
00463     VRType vr388;
00464     VRType vr389;
00465     VRType vr390;
00466     VRType vr391;
00467     VRType vr392;
00468     VRType vr393;
00469     VRType vr394;
00470     VRType vr395;
00471     VRType vr396;
00472     VRType vr397;
00473     VRType vr398;
00474     VRType vr399;
00475     VRType vr400;
00476     VRType vr401;
00477     VRType vr402;
00478     VRType vr403;
00479     VRType vr404;
00480     VRType vr405;
00481     VRType vr406;
00482     VRType vr407;
00483     VRType vr408;
00484     VRType vr409;
00485     VRType vr410;
00486     VRType vr411;
00487     VRType vr412;
00488     VRType vr413;
00489     VRType vr414;
00490     VRType vr415;
00491     VRType vr416;
00492     VRType vr417;
00493     VRType vr418;
00494     VRType vr419;
00495     VRType vr420;
00496     VRType vr421;
00497     VRType vr422;
00498     VRType vr423;
00499     VRType vr424;
00500     VRType vr425;
00501     VRType vr426;
00502     VRType vr427;
00503     VRType vr428;
00504     VRType vr429;
00505     VRType vr430;
00506     VRType vr431;
00507     VRType vr432;
00508     VRType vr433;
00509     VRType vr434;
00510     VRType vr435;
00511     VRType vr436;
00512     VRType vr437;
00513     VRType vr438;
00514     VRType vr439;
00515     VRType vr440;
00516     VRType vr441;
00517     VRType vr442;
00518     VRType vr443;
00519     VRType vr444;
00520     VRType vr445;
00521     VRType vr446;
00522     VRType vr447;
00523     VRType vr448;
00524     VRType vr449;
00525     VRType vr450;
00526     VRType vr451;
00527     VRType vr452;
00528     VRType vr453;
00529     VRType vr454;
00530     VRType vr455;
00531     VRType vr456;
00532     VRType vr457;
00533     VRType vr458;
00534     VRType vr459;
00535     VRType vr460;
00536     VRType vr461;
00537     VRType vr462;
00538     VRType vr463;
00539     VRType vr464;
00540     VRType vr465;
00541     VRType vr466;
00542     VRType vr467;
00543     VRType vr468;
00544     VRType vr469;
00545     VRType vr470;
00546     VRType vr471;
00547     VRType vr472;
00548     VRType vr473;
00549     VRType vr474;
00550     VRType vr475;
00551     VRType vr476;
00552     VRType vr477;
00553     VRType vr478;
00554     VRType vr479;
00555     VRType vr480;
00556     VRType vr481;
00557     VRType vr482;
00558     VRType vr483;
00559     VRType vr484;
00560     VRType vr485;
00561     VRType vr486;
00562     VRType vr487;
00563     VRType vr488;
00564     VRType vr489;
00565     VRType vr490;
00566     VRType vr491;
00567     VRType vr492;
00568     VRType vr493;
00569     VRType vr494;
00570     VRType vr495;
00571     VRType vr496;
00572     VRType vr497;
00573     VRType vr498;
00574     VRType vr499;
00575     VRType vr500;
00576     VRType vr501;
00577     VRType vr502;
00578     VRType vr503;
00579     VRType vr504;
00580     VRType vr505;
00581     VRType vr506;
00582     VRType vr507;
00583     VRType vr508;
00584     VRType vr509;
00585     VRType vr510;
00586     VRType vr511;
00587     VRType vr512;
00588     VRType vr513;
00589     VRType vr514;
00590     VRType vr515;
00591     VRType vr516;
00592     VRType vr517;
00593     VRType vr518;
00594     VRType vr519;
00595     VRType vr520;
00596     VRType vr521;
00597     VRType vr522;
00598     VRType vr523;
00599     VRType vr524;
00600     VRType vr525;
00601     VRType vr526;
00602     VRType vr527;
00603     VRType vr528;
00604     VRType vr529;
00605     VRType vr530;
00606     VRType vr531;
00607     VRType vr532;
00608     VRType vr533;
00609     VRType vr534;
00610     VRType vr535;
00611     VRType vr536;
00612     VRType vr537;
00613     VRType vr538;
00614     VRType vr539;
00615     VRType vr540;
00616     VRType vr541;
00617     VRType vr542;
00618     VRType vr543;
00619     VRType vr544;
00620     VRType vr545;
00621     VRType vr546;
00622     VRType vr547;
00623     VRType vr548;
00624     VRType vr549;
00625     VRType vr550;
00626     VRType vr551;
00627     VRType vr552;
00628     VRType vr553;
00629     VRType vr554;
00630     VRType vr555;
00631     VRType vr556;
00632     VRType vr557;
00633     VRType vr558;
00634     VRType vr559;
00635     VRType vr560;
00636     VRType vr561;
00637     VRType vr562;
00638     VRType vr563;
00639     VRType vr564;
00640     VRType vr565;
00641     VRType vr566;
00642     VRType vr567;
00643     VRType vr568;
00644     VRType vr569;
00645     VRType vr570;
00646     VRType vr571;
00647     VRType vr572;
00648     VRType vr573;
00649     VRType vr574;
00650     VRType vr575;
00651     VRType vr576;
00652     VRType vr577;
00653     VRType vr578;
00654     VRType vr579;
00655     VRType vr580;
00656     VRType vr581;
00657     VRType vr582;
00658     VRType vr583;
00659     VRType vr584;
00660     VRType vr585;
00661     VRType vr586;
00662     VRType vr587;
00663     VRType vr588;
00664     VRType vr589;
00665     VRType vr590;
00666     VRType vr591;
00667     VRType vr592;
00668     VRType vr593;
00669     VRType vr594;
00670     VRType vr595;
00671     VRType vr596;
00672     VRType vr597;
00673     VRType vr598;
00674     VRType vr599;
00675     VRType vr600;
00676     VRType vr601;
00677     VRType vr602;
00678     VRType vr603;
00679     VRType vr604;
00680     VRType vr605;
00681     VRType vr606;
00682     VRType vr607;
00683     VRType vr608;
00684     VRType vr609;
00685     VRType vr610;
00686     VRType vr611;
00687     VRType vr612;
00688     VRType vr613;
00689     VRType vr614;
00690     VRType vr615;
00691     VRType vr616;
00692     VRType vr617;
00693     VRType vr618;
00694     VRType vr619;
00695     VRType vr620;
00696     VRType vr621;
00697     VRType vr622;
00698     VRType vr623;
00699     VRType vr624;
00700     VRType vr625;
00701     VRType vr626;
00702     VRType vr627;
00703     VRType vr628;
00704     VRType vr629;
00705     VRType vr630;
00706     VRType vr631;
00707     VRType vr632;
00708     VRType vr633;
00709     VRType vr634;
00710     VRType vr635;
00711     VRType vr636;
00712     VRType vr637;
00713     VRType vr638;
00714     VRType vr639;
00715     VRType vr640;
00716     VRType vr641;
00717     VRType vr642;
00718     VRType vr643;
00719     VRType vr644;
00720     VRType vr645;
00721     VRType vr646;
00722     VRType vr647;
00723     VRType vr648;
00724     VRType vr649;
00725     VRType vr650;
00726     VRType vr651;
00727     VRType vr652;
00728     VRType vr653;
00729     VRType vr654;
00730     VRType vr655;
00731     VRType vr656;
00732     VRType vr657;
00733     VRType vr658;
00734     VRType vr659;
00735     VRType vr660;
00736     VRType vr661;
00737     VRType vr662;
00738     VRType vr663;
00739     VRType vr664;
00740     VRType vr665;
00741     VRType vr666;
00742     VRType vr667;
00743     VRType vr668;
00744     VRType vr669;
00745     VRType vr670;
00746     VRType vr671;
00747     VRType vr672;
00748     VRType vr673;
00749     VRType vr674;
00750     VRType vr675;
00751     VRType vr676;
00752     VRType vr677;
00753     VRType vr678;
00754     VRType vr679;
00755     VRType vr680;
00756     VRType vr681;
00757     VRType vr682;
00758     VRType vr683;
00759     VRType vr684;
00760     VRType vr685;
00761     VRType vr686;
00762     VRType vr687;
00763     VRType vr688;
00764     VRType vr689;
00765     VRType vr690;
00766     VRType vr691;
00767     VRType vr692;
00768     VRType vr693;
00769     VRType vr694;
00770     VRType vr695;
00771     VRType vr696;
00772     VRType vr697;
00773     VRType vr698;
00774     VRType vr699;
00775     VRType vr700;
00776     VRType vr701;
00777     VRType vr702;
00778     VRType vr703;
00779     VRType vr704;
00780     VRType vr705;
00781     VRType vr706;
00782     VRType vr707;
00783     VRType vr708;
00784     VRType vr709;
00785     VRType vr710;
00786     VRType vr711;
00787     VRType vr712;
00788     VRType vr713;
00789     VRType vr714;
00790     VRType vr715;
00791     VRType vr716;
00792     VRType vr717;
00793     VRType vr718;
00794     VRType vr719;
00795     VRType vr720;
00796     VRType vr721;
00797     VRType vr722;
00798     VRType vr723;
00799     VRType vr724;
00800     VRType vr725;
00801     VRType vr726;
00802     VRType vr727;
00803     VRType vr728;
00804     VRType vr729;
00805     VRType vr730;
00806     VRType vr731;
00807     VRType vr732;
00808     VRType vr733;
00809     VRType vr734;
00810     VRType vr735;
00811     VRType vr736;
00812     VRType vr737;
00813     VRType vr738;
00814     VRType vr739;
00815     VRType vr740;
00816     VRType vr741;
00817     VRType vr742;
00818     VRType vr743;
00819     VRType vr744;
00820     VRType vr745;
00821     VRType vr746;
00822     VRType vr747;
00823     VRType vr748;
00824     VRType vr749;
00825     VRType vr750;
00826     VRType vr751;
00827     VRType vr752;
00828     VRType vr753;
00829     VRType vr754;
00830     VRType vr755;
00831     VRType vr756;
00832     VRType vr757;
00833     VRType vr758;
00834     VRType vr759;
00835     VRType vr760;
00836     VRType vr761;
00837     VRType vr762;
00838     VRType vr763;
00839     VRType vr764;
00840     VRType vr765;
00841     VRType vr766;
00842     VRType vr767;
00843     VRType vr768;
00844     VRType vr769;
00845     VRType vr770;
00846     VRType vr771;
00847     VRType vr772;
00848     VRType vr773;
00849     VRType vr774;
00850     VRType vr775;
00851     VRType vr776;
00852     VRType vr777;
00853     VRType vr778;
00854     VRType vr779;
00855     VRType vr780;
00856     VRType vr781;
00857     VRType vr782;
00858     VRType vr783;
00859     VRType vr784;
00860     VRType vr785;
00861     VRType vr786;
00862     VRType vr787;
00863     VRType vr788;
00864     VRType vr789;
00865     VRType vr790;
00866     VRType vr791;
00867     VRType vr792;
00868     VRType vr793;
00869     VRType vr794;
00870     VRType vr795;
00871     VRType vr796;
00872     VRType vr797;
00873     VRType vr798;
00874     VRType vr799;
00875     VRType vr800;
00876     VRType vr801;
00877     VRType vr802;
00878     VRType vr803;
00879     VRType vr804;
00880     VRType vr805;
```

```

00088     UL = 8388608,
00089     UN = 16777216,
00090     UR = 1073741824, // 2^30
00091     US = 33554432,
00092     UT = 67108864,
00093     UV = 8589934592, // 2^33
00094     OB_OW = OB | OW,
00095     US_SS = US | SS,
00096     US_SS_OW = US | SS | OW,
00097     US_OW = US | OW,
00098     // The following do not have a VRString equivalent (ie cannot be found in PS 3.6)
00099     VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US, // if( VR
& VL16 ) => VR has its VL coded over 16bits
00100     VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV, // if( VR & VL32 ) => VR has its VL coded
over 32bits
00101     VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT,
00102     VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV, // FIXME: UN ?
00103     // PS 3.5:
00104     // Data Elements with a VR of SQ, OD, OF, OL, OW, OB or UN shall always have a Value Multiplicity of one.
00105     // GDCM is adding a couple more: AS, LT, ST, UT
00106     VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN, // All those VR have a VM1
00107     VRALL = VRASCII | VRBINARY,
00108     VR_END = UV+1 // Invalid VR, need to be max(VRType)+1
00109 };
00110
00111     static const char *GetVRString(VRType vr);
00112
00113     // This function will only look at the very first two chars nothing else
00114     static VRType GetTypeFromFile(const char *vr);
00115
00116     // You need to make sure end of string is \0
00117     static VRType GetType(const char *vr);
00118     static const char *GetStringFromFile(VRType vr);
00119
00120     static bool IsValid(const char *vr);
00121     // Check if vr1 is valid against vr2,
00122     // Typically vr1 is read from the file and vr2 is taken from the dict
00123     static bool IsValid(const char *vr1, VRType vr2);
00124     //static bool IsValid(const VRType &vr1, const VRType &vr2);
00125     // Find out if the string read is byte swapped
00126     static bool IsSwap(const char *vr);
00127
00128     // Size read on disk
00129     // FIXME: int ?
00130     int GetLength() const {
00131         return VR::GetLength(VRField);
00132     }
00133     unsigned int GetSizeof() const;
00134     static uint32_t GetLength(VRType vr) {
00135         //if( vr == VR::INVALID ) return 4;
00136         if( vr & VL32 )
00137         {
00138             return 4;
00139         }
00140         else
00141             return 2;
00142     }
00143
00144     // Some use of template metaprograming with ugly macro
00145     static bool IsBinary(VRType vr);
00146     static bool IsASCII(VRType vr);
00147     // TODO: REMOVE ME
00148     static bool CanDisplay(VRType vr);
00149     // TODO: REMOVE ME
00150     static bool IsBinary2(VRType vr);
00151     // TODO: REMOVE ME
00152     static bool IsASCII2(VRType vr);
00153
00154     VR(VRType vr = INVALID):VRField(vr) { }
00155     //VR(VR const &vr):VRField(vr.VRField) { }
00156     std::istream &Read(std::istream &is)
00157     {
00158         char vr[2];
00159         is.read(vr, 2);
00160         VRField = GetTypeFromFile(vr);
00161         gdcm_assert( VRField != VR::VR_END );
00162         if( VRField == VR::INVALID )
00163         {
00164             // \0\2 Data/TherapysGDCM120Bug.dcm
00165             // \0\0
Data/MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm

```

```

00166 // \0\4 Data/BugGDCM2_UndefItemWrongVL.dcm
00167 // \44\0 Data/gdcm-MR-PHILIPS-16-Multi-Seq.dcm
00168 // \0\20 Data/ExplicitVRforPublicElementsImplicitVRforShadowElements.dcm
00169 // \0\3 Data/DMCPACS_ExplicitImplicit_BogusIOP.dcm
00170 // \0\4 Data/THERALYS-12-MONO2-Uncompressed-Even_Length_Tag.dcm
00171 // \0\4 Data/PrivateGEImplicitVRBigEndianTransferSyntax16Bits.dcm
00172 // \0\4 Data/GE_DLX-8-MONO2-PrivateSyntax.dcm
00173 throw Exception( "INVALID VR" );
00174 }
00175 if( VRField & VL32 )
00176 {
00177 #if 0
00178 // For some reason this seems slower on my linux box...
00179 is.seekg(2, std::ios::cur );
00180 #else
00181 char dumb[2];
00182 is.read(dumb, 2);
00183 if( !(dumb[0] == 0 && dumb[1] == 0 ))
00184 {
00185 // JDDICOM_Sample4.dcm
00186 gdcmDebugMacro( "32bits VR contains non zero bytes. Skipped" );
00187 }
00188 #endif
00189 }
00190 return is;
00191 }
00192
00193 const std::ostream &Write(std::ostream &os) const
00194 {
00195 VRType vrfld = VRField;
00196 gdcmAssertAlwaysMacro( !IsDual() );
00197 if( vrfld == VR::INVALID )
00198 {
00199 //vrfld = VR::UN;
00200 }
00201 const char *vr = GetVRString(vrfld);
00202 //gdcm_assert( strlen( vr ) == 2 );
00203 gdcm_assert( vr[0] && vr[1] && vr[2] == 0 );
00204 os.write(vr, 2);
00205 // See PS 3.5, Data Element Structure With Explicit VR
00206 if( vrfld & VL32 )
00207 {
00208 const char dumb[2] = {0, 0};
00209 os.write(dumb,2);
00210 }
00211 return os;
00212 }
00213 friend std::ostream &operator«(std::ostream &os, const VR &vr);
00214
00215 operator VRType () const { return VRField; }
00216
00217 unsigned int GetSize() const;
00218
00219 bool Compatible(VR const &vr) const;
00220
00221 bool IsVRFile() const;
00222
00223 bool IsDual() const;
00224
00225 private:
00226 // Internal function that map a VRType to an index in the VRStrings table
00227 static unsigned int GetIndex(VRType vr);
00228 VRType VRField;
00229 };
00230 //-----
00231 inline std::ostream &operator«(std::ostream &_os, const VR &val)
00232 {
00233 //_os « VR::GetVRStringFromFile(val.VRField);
00234 _os « VR::GetVRString(val.VRField);
00235 return _os;
00236 }
00237
00238 // Apparently SWIG is not happy with something, somewhere below...
00239 #ifndef SWIG
00240
00241 // Tells whether VR Type is ASCII or Binary
00242 template<long long T> struct VRToEncoding;
00243 // Convert from VR Type to real underlying type
00244 template<long long T> struct VRToType;
00245 #define TYPETOENCODING(type,rep, rtype) \
00246 template<> struct VRToEncoding<VR::type> \

```

```

00247 { enum:long long { Mode = VR::rep }; }; \
00248 template<> struct VRToType<VR::type> \
00249 { typedef rtype Type; };
00250
00251
00252 // Do not use me
00253 struct UI { char Internal[64+1];
00254 friend std::ostream& operator«(std::ostream &_os, const UI &_val);
00255 };
00256 inline std::ostream& operator«(std::ostream &_os, const UI &_val)
00257 {
00258     _os « _val.Internal;
00259     return _os;
00260 }
00261
00262 typedef String<'\',16> AECComp;
00263 typedef String<'\',64> ASCComp;
00264 typedef String<'\',16> CSCComp;
00265 typedef String<'\',64> DACComp;
00266 typedef String<'\',64> DTCComp;
00267 typedef String<'\',64> LOComp;
00268 typedef String<'\',64> LTComp;
00269 typedef String<'\',64> PNComp;
00270 typedef String<'\',64> SHComp;
00271 typedef String<'\',64> STComp;
00272 typedef String<'\',4294967294> UCComp;
00273 typedef String<'\',4294967294> URComp;
00274 typedef String<'\',16> TMComp;
00275 typedef String<'\',64,0> UIComp;
00276 typedef String<'\',64> UTCComp;
00277
00278
00279 // TODO: Could be generated from XML file
00280 TYPETOENCODING(AE,VRASCII ,AECComp)
00281 TYPETOENCODING(AS,VRASCII ,ASCComp)
00282 TYPETOENCODING(AT,VRBINARY,Tag)
00283 TYPETOENCODING(CS,VRASCII ,CSCComp)
00284 TYPETOENCODING(DA,VRASCII ,DACComp)
00285 TYPETOENCODING(DS,VRASCII ,double)
00286 TYPETOENCODING(DT,VRASCII ,DTCComp)
00287 TYPETOENCODING(FL,VRBINARY,float)
00288 TYPETOENCODING(FD,VRBINARY,double)
00289 TYPETOENCODING(IS,VRASCII ,int32_t)
00290 TYPETOENCODING(LO,VRASCII ,LOComp)
00291 TYPETOENCODING(LT,VRASCII ,LTComp)
00292 TYPETOENCODING(OB,VRBINARY,uint8_t)
00293 TYPETOENCODING(OD,VRBINARY,double)
00294 TYPETOENCODING(OF,VRBINARY,float)
00295 TYPETOENCODING(OL,VRBINARY,uint32_t)
00296 TYPETOENCODING(OV,VRBINARY,uint64_t)
00297 TYPETOENCODING(OW,VRBINARY,uint16_t)
00298 TYPETOENCODING(PN,VRASCII ,PNComp)
00299 TYPETOENCODING(SH,VRASCII ,SHComp)
00300 TYPETOENCODING(SL,VRBINARY,int32_t)
00301 TYPETOENCODING(SQ,VRBINARY,unsigned char) // FIXME
00302 TYPETOENCODING(SS,VRBINARY,int16_t)
00303 TYPETOENCODING(ST,VRASCII ,STComp)
00304 TYPETOENCODING(SV,VRBINARY,int64_t)
00305 TYPETOENCODING(TM,VRASCII ,TMComp)
00306 TYPETOENCODING(UC,VRASCII ,UCComp)
00307 TYPETOENCODING(UI,VRASCII ,UIComp)
00308 TYPETOENCODING(UL,VRBINARY,uint32_t)
00309 TYPETOENCODING(UN,VRBINARY,uint8_t) // FIXME ?
00310 TYPETOENCODING(UR,VRASCII ,URComp)
00311 TYPETOENCODING(US,VRBINARY,uint16_t)
00312 TYPETOENCODING(UT,VRASCII ,UTCComp)
00313 TYPETOENCODING(UV,VRBINARY,uint64_t)
00314
00315 #define VRTypeTemplateCase(type) \
00316     case VR::type: \
00317         return sizeof ( VRToType<VR::type>::Type );
00318
00319 inline unsigned int VR::GetSize() const
00320 {
00321     switch(VRField)
00322     {
00323         VRTypeTemplateCase(AE)
00324         VRTypeTemplateCase(AS)
00325         VRTypeTemplateCase(AT)
00326         VRTypeTemplateCase(CS)
00327         VRTypeTemplateCase(DA)

```

```

00328 VRTypeTemplateCase(DS)
00329 VRTypeTemplateCase(DT)
00330 VRTypeTemplateCase(FL)
00331 VRTypeTemplateCase(FD)
00332 VRTypeTemplateCase(IS)
00333 VRTypeTemplateCase(LO)
00334 VRTypeTemplateCase(LT)
00335 VRTypeTemplateCase(OB)
00336 VRTypeTemplateCase(OD)
00337 VRTypeTemplateCase(OF)
00338 VRTypeTemplateCase(OL)
00339 VRTypeTemplateCase(OV)
00340 VRTypeTemplateCase(OW)
00341 VRTypeTemplateCase(PN)
00342 VRTypeTemplateCase(SH)
00343 VRTypeTemplateCase(SL)
00344 VRTypeTemplateCase(SQ)
00345 VRTypeTemplateCase(SS)
00346 VRTypeTemplateCase(ST)
00347 VRTypeTemplateCase(SV)
00348 VRTypeTemplateCase(TM)
00349 VRTypeTemplateCase(UC)
00350 VRTypeTemplateCase(UI)
00351 VRTypeTemplateCase(UL)
00352 VRTypeTemplateCase(UN)
00353 VRTypeTemplateCase(UR)
00354 VRTypeTemplateCase(US)
00355 VRTypeTemplateCase(UT)
00356 VRTypeTemplateCase(UV)
00357 case VR::US_SS:
00358     return 2;
00359
00360 case VR::INVALID:
00361 case VR::OB_OW:
00362 case VR::US_SS_OW:
00363 case VR::US_OW:
00364 case VR::VL16:
00365 case VR::VL32:
00366 case VR::VRASCII:
00367 case VR::VRBINARY:
00368 case VR::VR_VM1:
00369 case VR::VRALL:
00370 case VR::VR_END:
00371 default:
00372     gdcmm_assert( 0 && "should not" );
00373 }
00374 return 0;
00375 }
00376 #endif // SWIG
00377
00378
00379 } // end namespace gdcmm
00380
00381 #endif //GDCMMVR_H

```

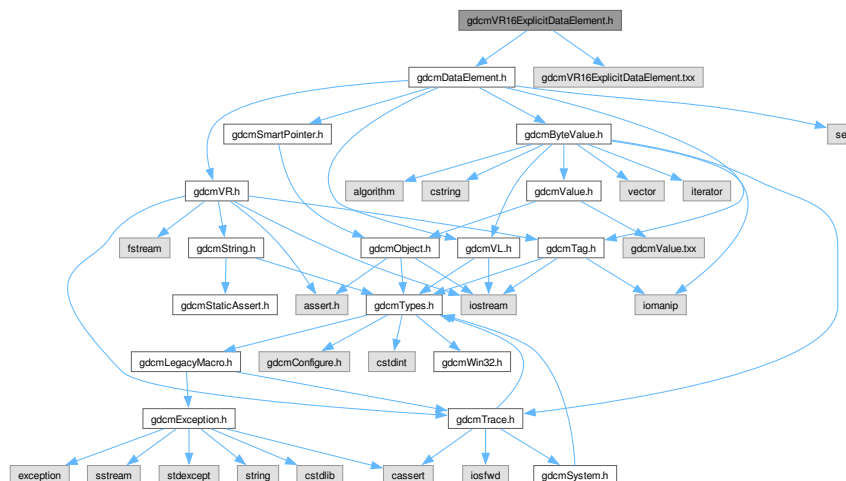
13.195 gdcmmVR16ExplicitDataElement.h File Reference

```

#include "gdcmmDataElement.h"
#include "gdcmmVR16ExplicitDataElement.txx"

```


Include dependency graph for gdcVR16ExplicitDataElement.h:



Classes

- class [gdc::VR16ExplicitDataElement](#)
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Namespaces

- namespace [gdc](#)

13.196 gdcVR16ExplicitDataElement.h

[Go to the documentation of this file.](#)

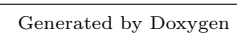
```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMVR16EXPLICITDATAELEMENT_H
00015  #define GDCMVR16EXPLICITDATAELEMENT_H
00016
00017  #include "gdcDataElement.h"
00018
00019  namespace gdc
00020  {
00021  // Data Element (Explicit)

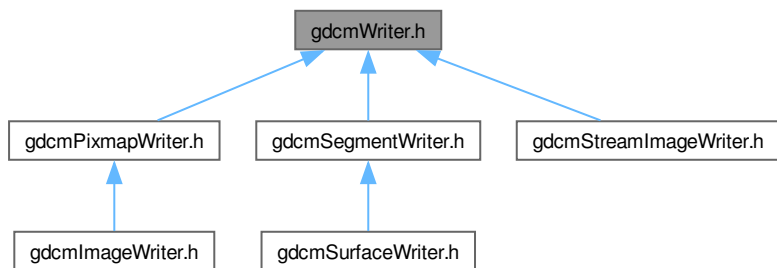
```

13.197 gdcmWriter.h File Reference

Include dependency graph for gdcMWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Writer](#)
[Writer](#) ala DOM (Document [Object](#) Model).

Namespaces

- namespace [gdcm](#)

13.198 gdcmWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014
00015  #ifndef GDCMWRITER_H
00016  #define GDCMWRITER_H
00017
00018  #include "gdcmFile.h"
00019
00020  namespace gdcm
00021  {
00022
00023  class FileMetaInformation;
00048  class GDCM_EXPORT Writer
00049  {
00050  public:
00051    Writer();
00052    virtual ~Writer();
  
```

```

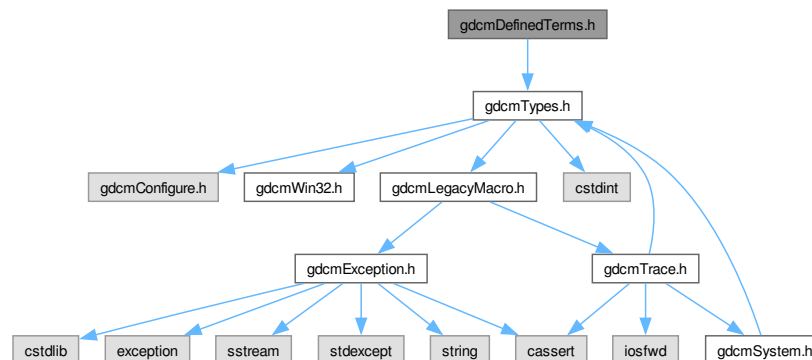
00053
00055 virtual bool Write(); // Execute()
00056
00058 void SetFileName(const char *filename_native);
00059
00061 void SetStream(std::ostream &output_stream) {
00062     Stream = &output_stream;
00063 }
00064
00066 void SetFile(const File& f) { F = f; }
00067 File &GetFile() { return *F; }
00068
00070 void SetCheckFileMetaInformation(bool b) { CheckFileMetaInformation = b; }
00071 void CheckFileMetaInformationOff() { CheckFileMetaInformation = false; }
00072 void CheckFileMetaInformationOn() { CheckFileMetaInformation = true; }
00073
00074 protected:
00075 void SetWriteDataSetOnly(bool b) { WriteDataSetOnly = b; }
00076
00077 protected:
00078 friend class StreamImageWriter;
00079 //this function is added for the StreamImageWriter, which needs to write
00080 //up to the pixel data and then stops right before writing the pixel data.
00081 //after that, for the raw codec at least, zeros are written for the length of the data
00082 std::ostream* GetStreamPtr() const { return Stream; }
00083
00084 protected:
00085 std::ostream *Stream;
00086 std::ofstream *Ofstream;
00087 bool GetCheckFileMetaInformation() const { return CheckFileMetaInformation; }
00088
00089 private:
00090 SmartPointer<File> F;
00091 bool CheckFileMetaInformation;
00092 bool WriteDataSetOnly;
00093 };
00094
00095 } // end namespace gdcM
00096
00097 #endif //GDCMWRITER_H

```

13.199 gdcMDefinedTerms.h File Reference

#include "gdcMTypes.h"

Include dependency graph for gdcMDefinedTerms.h:



Classes

- class [gdcM::DefinedTerms](#)

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data [Element](#) with Defined Terms that does not contain a [Value](#) equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation [Type](#) ID (4008,0210) is an example of a Data [Element](#) having Defined Terms. It is defined to have a [Value](#) that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data [Element](#) has Defined Terms other Interpretation [Type](#) IDs may be defined by the implementor.

Namespaces

- namespace [gdcml](#)

13.200 gdcmlDefinedTerms.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDEFINEDTERMS_H
00015 #define GDCMDEFINEDTERMS_H
00016
00017 #include "gdcmlTypes.h"
00018
00019 namespace gdcml
00020 {
00021     class GDCM_EXPORT DefinedTerms
00022     {
00023     public:
00024         DefinedTerms() = default;
00025     private:
00026     };
00027 } // end namespace gdcml
00028
00029 #endif //GDCMDEFINEDTERMS_H

```

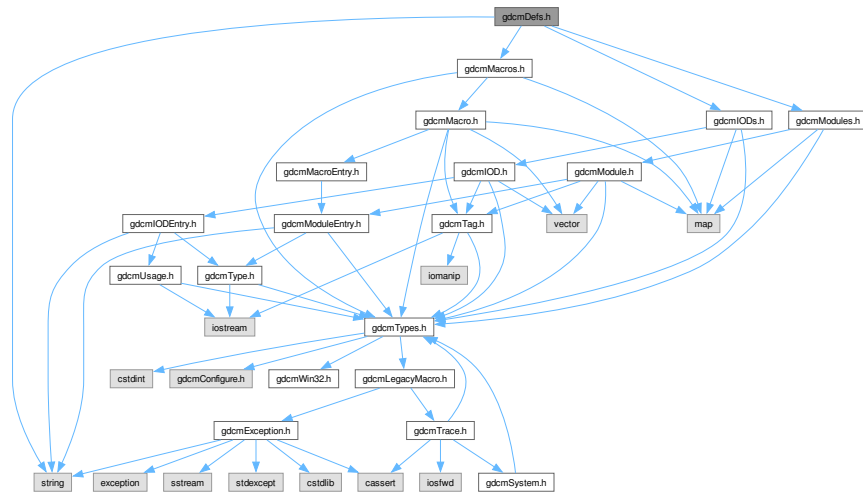
13.201 gdcmlDefs.h File Reference

```

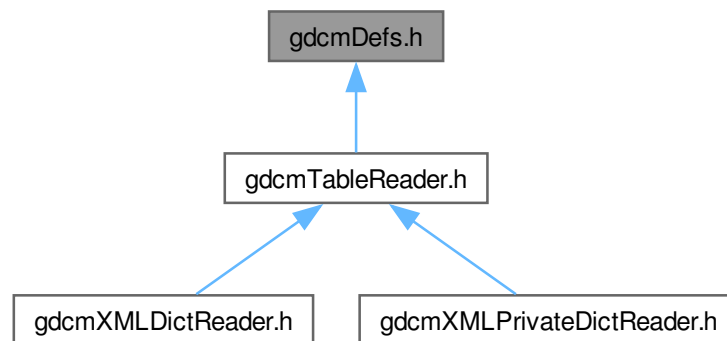
#include "gdcmlModules.h"
#include "gdcmlMacros.h"
#include "gdcmlIODs.h"

```

Include dependency graph for gdcMDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::Defs`
 FIXME I do not like the name '`Defs`'.

Namespaces

- namespace `gdcm`

13.202 gdcmDefs.h

[Go to the documentation of this file.](#)

```

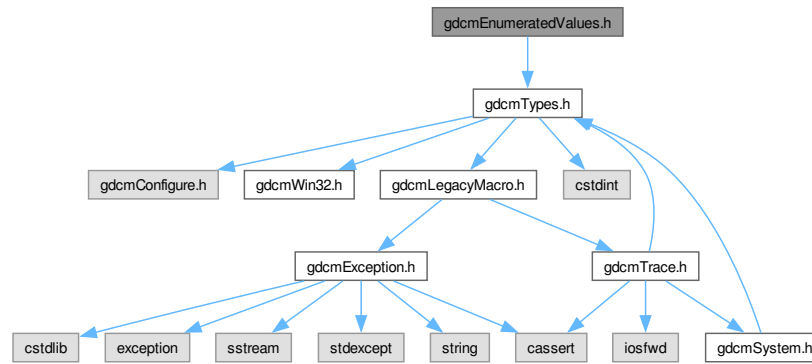
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDEFS_H
00015 #define GDCMDEFS_H
00016
00017 #include "gdcmModules.h"
00018 #include "gdcmMacros.h"
00019 #include "gdcmIODs.h"
00020
00021 #include <string>
00022
00023 namespace gdcm
00024 {
00025 class DataSet;
00026 class File;
00027 class MediaStorage;
00032 class GDCM_EXPORT Defs
00033 {
00034 public:
00035 Defs();
00036 ~Defs();
00037 Defs &operator=(const Defs &val) = delete;
00038 Defs(const Defs &val) = delete;
00039
00040 const Modules &GetModules() const { return Part3Modules; }
00041 Modules &GetModules() { return Part3Modules; }
00042
00043 const Macros &GetMacros() const { return Part3Macros; }
00044 Macros &GetMacros() { return Part3Macros; }
00045
00046 const IODs & GetIODs() const { return Part3IODs; }
00047 IODs & GetIODs() { return Part3IODs; }
00050
00051 bool IsEmpty() const { return GetModules().IsEmpty(); }
00052
00053 bool Verify(const File& file) const;
00054
00055 // \deprecated DO NOT USE
00056 bool Verify(const DataSet& ds) const;
00057
00058 Type GetTypeFromTag(const File& file, const Tag& tag) const;
00059
00060 static const char *GetIODNameFromMediaStorage(MediaStorage const &ms);
00061
00062 const IOD& GetIODFromFile(const File& file) const;
00063
00064 protected:
00065 friend class Global;
00066 void LoadDefaults();
00067 void LoadFromFile(const char *filename);
00068
00069 private:
00070 // Part 3 stuff:
00071 Macros Part3Macros;
00072 Modules Part3Modules;
00073 IODs Part3IODs;
00074
00075 };
00076
00077
00078 } // end namespace gdcm
00079
00080 #endif //GDCMDEFS_H

```

13.203 gdcmEnumeratedValues.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmEnumeratedValues.h:



Classes

- class [gdcm::EnumeratedValues](#)

Element. A Data [Element](#) with Enumerated Values that does not have a [Value](#) equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

Namespaces

- namespace [gdcm](#)

13.204 gdcmEnumeratedValues.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMENUMERATEDVALUES_H
00015  #define GDCMENUMERATEDVALUES_H
00016
00017  #include "gdcmTypes.h"

```



```

00018
00019 namespace gdcm
00020 {
00034 class GDCM_EXPORT EnumeratedValues
00035 {
00036 public:
00037     EnumeratedValues() = default;
00038 private:
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMENUMERATEDVALUES_H

```

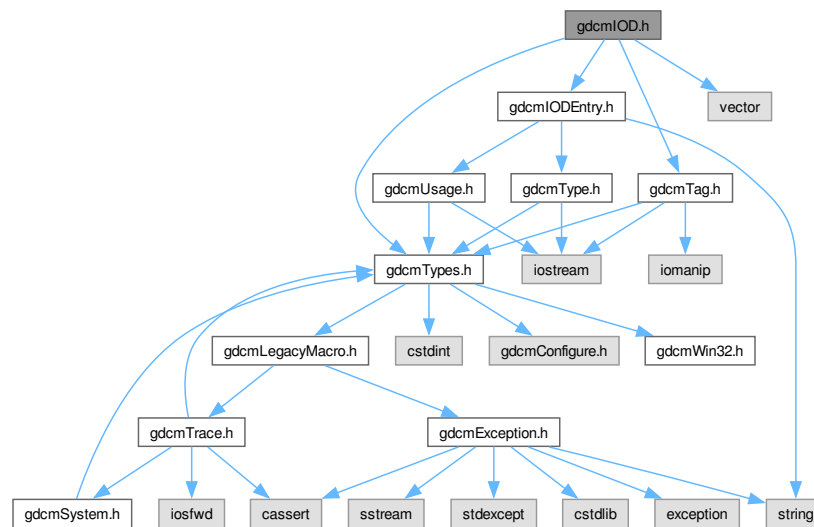
13.205 gdcmIOD.h File Reference

```

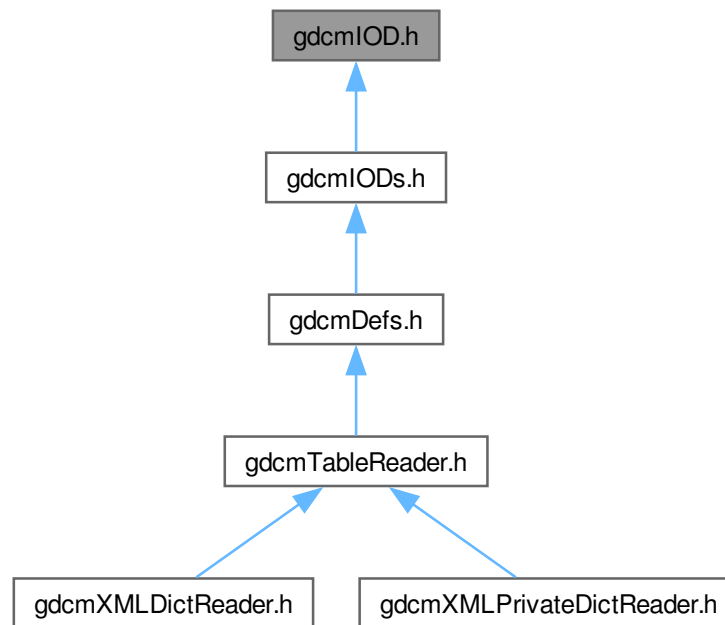
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"
#include <vector>

```

Include dependency graph for gdcmIOD.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::IOD](#)
Class for representing a [IOD](#).

Namespaces

- namespace [gdcml](#)

Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IOD &_val)`

13.206 gdcmlOD.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
```

```

00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMIOD_H
00015 #define GDCMIOD_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include "gdcmIODEntry.h"
00020
00021 #include <vector>
00022
00023 namespace gdcm
00024 {
00025 class DataSet;
00026 class Defs;
00027
00034 class GDCM_EXPORT IOD
00035 {
00036 public:
00037 typedef std::vector<IODEntry> MapIODEntry;
00038 typedef MapIODEntry::size_type SizeType;
00039
00040 IOD() = default;
00041 friend std::ostream& operator<<(std::ostream& _os, const IOD &_val);
00042
00043 void Clear() { IODInternal.clear(); }
00044
00045 void AddIODEntry(const IODEntry &iode)
00046 {
00047 IODInternal.push_back(iode);
00048 }
00049
00050 SizeType GetNumberOfIODs() const {
00051 return IODInternal.size();
00052 }
00053
00054 const IODEntry& GetIODEntry(SizeType idx) const
00055 {
00056 return IODInternal[idx];
00057 }
00058
00059 Type GetTypeFromTag(const Defs &defs, const Tag& tag) const;
00060
00061 private:
00062 //IOD &operator=(const IOD &_val); // purposely not implemented
00063 //IOD(const IOD &_val); // purposely not implemented
00064
00065 MapIODEntry IODInternal;
00066 };
00067 //-----
00068 inline std::ostream& operator<<(std::ostream& _os, const IOD &_val)
00069 {
00070 IOD::MapIODEntry::const_iterator it = _val.IODInternal.begin();
00071 for(;it != _val.IODInternal.end(); ++it)
00072 {
00073 _os << *it << '\n';
00074 }
00075
00076 return _os;
00077 }
00078
00079 } // end namespace gdcm
00080
00081 #endif //GDCMIOD_H

```

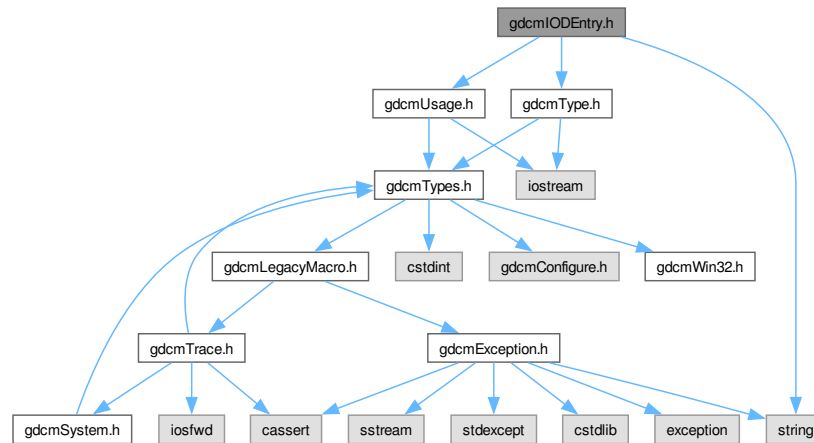
13.207 gdcmIODEntry.h File Reference

```
#include "gdcmUsage.h"
```

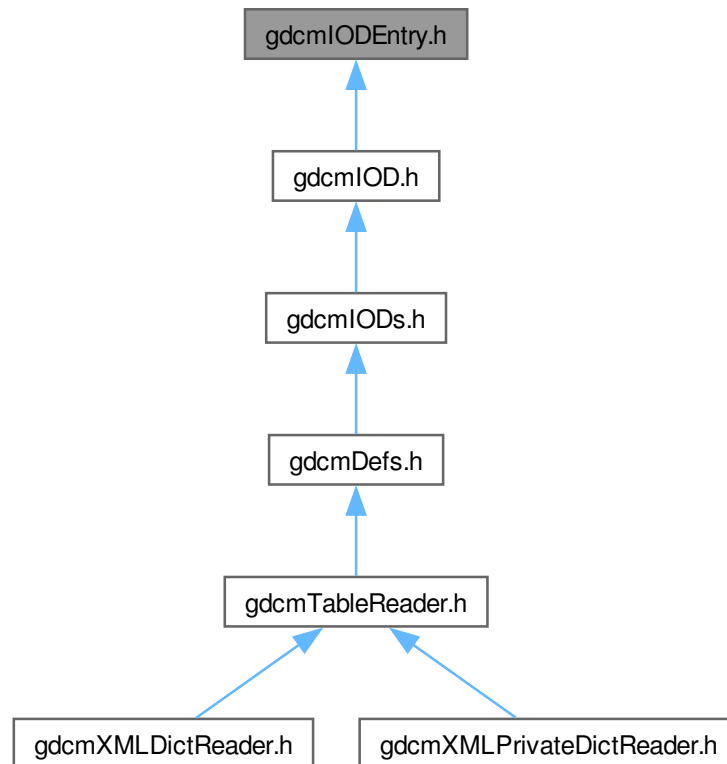
```
#include "gdcmType.h"
```

```
#include <string>
```

Include dependency graph for gdcmIODEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::IODEntry](#)
Class for representing a [IODEntry](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)`

13.208 gdcmIODEntry.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIODENTRY_H
00015 #define GDCMIODENTRY_H
00016
00017 #include "gdcmUsage.h"
00018 #include "gdcmType.h"
00019
00020 #include <string>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT IODEntry
00025     {
00026     public:
00027         IODEntry(const char *name = "", const char *ref = "", const char *inUsage = ""):Name(name),Ref(ref),usage(inUsage) {
00028         }
00029         friend std::ostream& operator<<(std::ostream& _os, const IODEntry &_val);
00030
00031         void SetIE(const char *ie) { IE = ie; }
00032         const char *GetIE() const { return IE.c_str(); }
00033
00034         void SetName(const char *name) { Name = name; }
00035         const char *GetName() const { return Name.c_str(); }
00036
00037         void SetRef(const char *ref) { Ref = ref; }
00038         const char *GetRef() const { return Ref.c_str(); }
00039
00040         void SetUsage(const char *inUsage) { usage = inUsage; }
00041         const char *GetUsage() const { return usage.c_str(); }
00042         Usage::UsageType GetUsageType() const;
00043
00044     private:
00045         std::string IE;
00046
00047         std::string Name;
00048
00049         std::string Ref;
00050
00051         std::string usage;
00052     };
00053
00054 //-----
00055 inline std::ostream& operator<<(std::ostream& _os, const IODEntry &_val)
00056 {
00057     _os << _val.IE << "\t" << _val.Name << "\t" << _val.Ref << "\t" << _val.usage;
00058     return _os;
00059 }
00060
00061 } // end namespace gdcm
00062
00063 #endif //GDCMIODENTRY_H

```

13.209 gdcmIODs.h File Reference

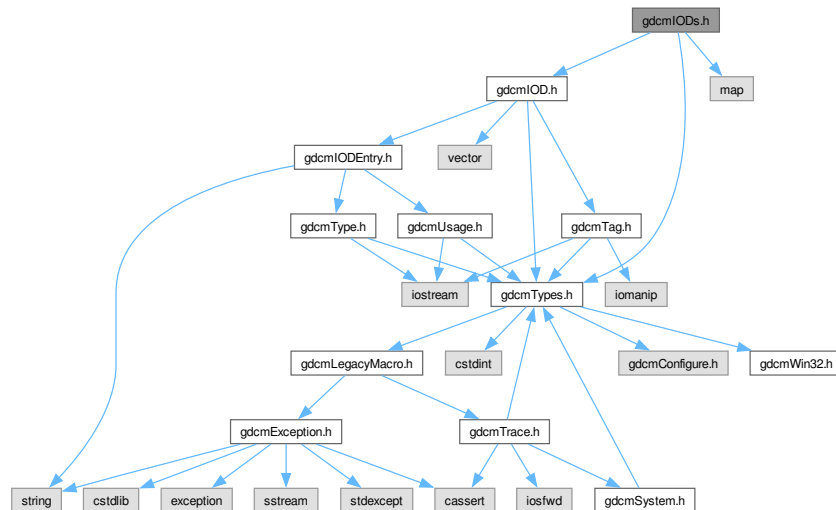
```

#include "gdcmTypes.h"
#include "gdcmIOD.h"

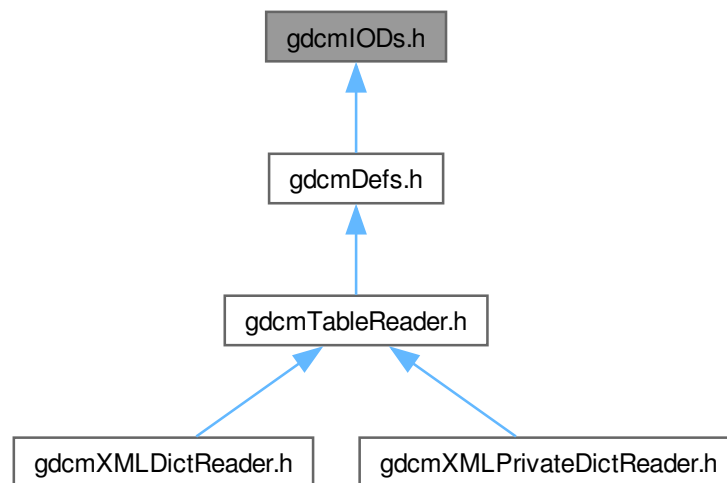
```

```
#include <map>
```

Include dependency graph for gdcmIODs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::IODs](#)
Class for representing a [IODs](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)`

13.210 [gdcmIODs.h](#)

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIODS_H
00015 #define GDCMIODS_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmIOD.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT IODs
00025     {
00026     public:
00027         typedef std::string IODName;
00028         typedef std::map<IODName, IOD> IODMapType;
00029
00030         IODs() = default;
00031         friend std::ostream& operator<<(std::ostream& _os, const IODs &_val);
00032
00033         void Clear() { IODsInternal.clear(); }
00034
00035         void AddIOD(const char *name, const IOD & module)
00036         {
00037             IODsInternal.insert(
00038                 IODMapType::value_type(name, module));
00039         }
00040         const IOD &GetIOD(const char *name) const
00041         {
00042             //return IODsInternal[name];
00043             IODMapType::const_iterator it = IODsInternal.find( name );
00044             gdcm_assert( it != IODsInternal.end() );
00045             gdcm_assert( it->first == name );
00046             return it->second;
00047         }
00048
00049         typedef IODMapType::const_iterator IODMapTypeConstIterator;
00050         IODMapTypeConstIterator Begin() const { return IODsInternal.begin(); }
00051         IODMapTypeConstIterator End() const { return IODsInternal.end(); }
00052
00053     private:
00054         IODMapType IODsInternal;
00055     };
00056
00057 //-----
00058 inline std::ostream& operator<<(std::ostream& _os, const IODs &_val)

```



```

00063 {
00064   IODs::IODMapType::const_iterator it = _val.IODsInternal.begin();
00065   for(; it != _val.IODsInternal.end(); ++it)
00066   {
00067     const std::string &name = it->first;
00068     const IOD &m = it->second;
00069     _os << name << " " << m << '\n';
00070   }
00071
00072   return _os;
00073 }
00074
00075
00076 } // end namespace gdcm
00077
00078 #endif //GDCMIODS_H

```

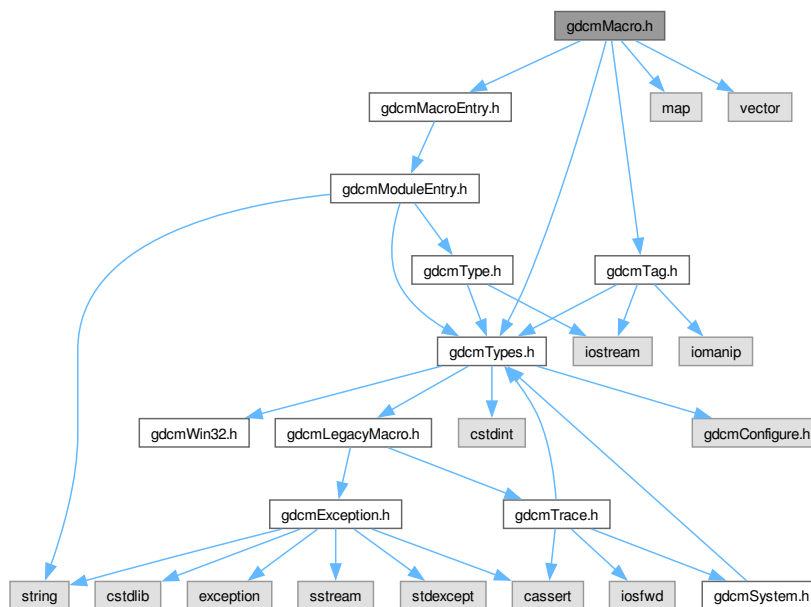
13.211 gdcmMacro.h File Reference

```

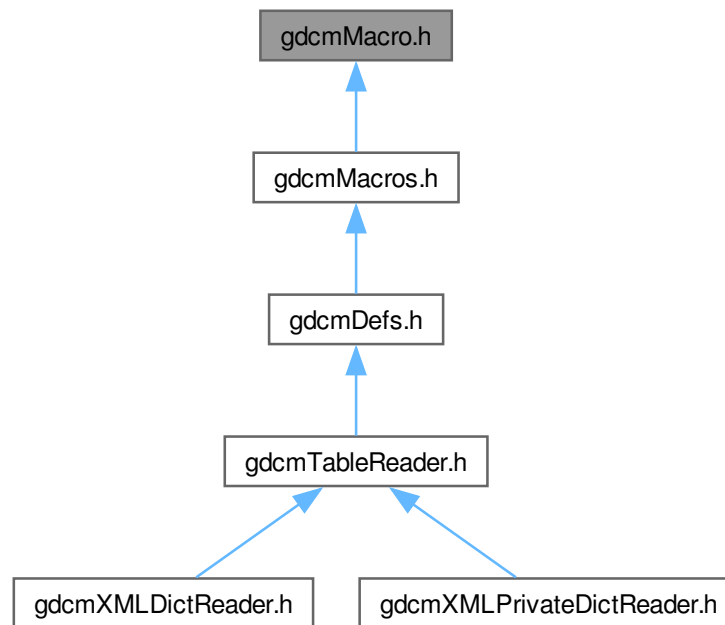
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>

```

Include dependency graph for gdcmMacro.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Macro](#)
Class for representing a [Macro](#).

Namespaces

- namespace [gdc](#)

Functions

- `std::ostream & gdc::operator<< (std::ostream &_os, const Macro &_val)`

13.212 gdcMacro.h

[Go to the documentation of this file.](#)

```
00001
00002
```

```
/*=====
```

```

00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcmmacro.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMMACRO_H
00015 #define GDCMMACRO_H
00016
00017 #include "gdcmmacroTypes.h"
00018 #include "gdcmmacroTag.h"
00019 #include "gdcmmacroMacroEntry.h"
00020
00021 #include <map>
00022 #include <vector>
00023
00024 namespace gdcmmacro
00025 {
00026
00027 class DataSet;
00028 class Usage;
00029 class GDCM_EXPORT Macro
00030 {
00031 public:
00032 typedef std::map<Tag, MacroEntry> MapModuleEntry;
00033 typedef std::vector<std::string> ArrayIncludeMacrosType;
00034
00035 //typedef MapModuleEntry::const_iterator ConstIterator;
00036 //typedef MapModuleEntry::iterator Iterator;
00037 //ConstIterator Begin() const { return ModuleInternal.begin(); }
00038 //Iterator Begin() { return ModuleInternal.begin(); }
00039 //ConstIterator End() const { return ModuleInternal.end(); }
00040 //Iterator End() { return ModuleInternal.end(); }
00041
00042 Macro() = default;
00043 friend std::ostream& operator<<(std::ostream& _os, const Macro& _val);
00044
00045 void Clear() { ModuleInternal.clear(); }
00046
00047 void AddMacroEntry(const Tag& tag, const MacroEntry & module)
00048 {
00049     ModuleInternal.insert(
00050         MapModuleEntry::value_type(tag, module));
00051 }
00052
00053 bool FindMacroEntry(const Tag &tag) const;
00054 const MacroEntry& GetMacroEntry(const Tag &tag) const;
00055
00056 void SetName(const char *name) { Name = name; }
00057 const char *GetName() const { return Name.c_str(); }
00058
00059 // Verify will print on std::cerr for error
00060 // Upon success will return true, false otherwise
00061 bool Verify(const DataSet& ds, Usage const & usage) const;
00062
00063 private:
00064 //Module &operator=(const Module &_val); // purposely not implemented
00065 //Module(const Module &_val); // purposely not implemented
00066
00067 MapModuleEntry ModuleInternal;
00068 std::string Name;
00069 };
00070 //-----
00071 inline std::ostream& operator<<(std::ostream& _os, const Macro &_val)
00072 {
00073     _os << _val.Name << '\n';
00074     Macro::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
00075     for(; it != _val.ModuleInternal.end(); ++it)
00076     {
00077         const Tag &t = it->first;
00078         const MacroEntry &de = it->second;
00079         _os << t << " " << de << '\n';
00080     }
00081     return _os;
00082 }

```

```

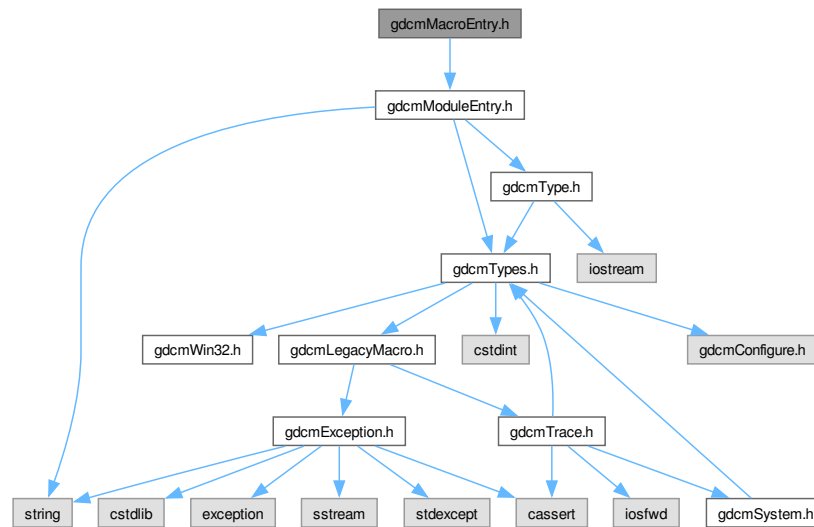
00093 }
00094
00095 } // end namespace gdcm
00096
00097 #endif //GDCMMACRO_H

```

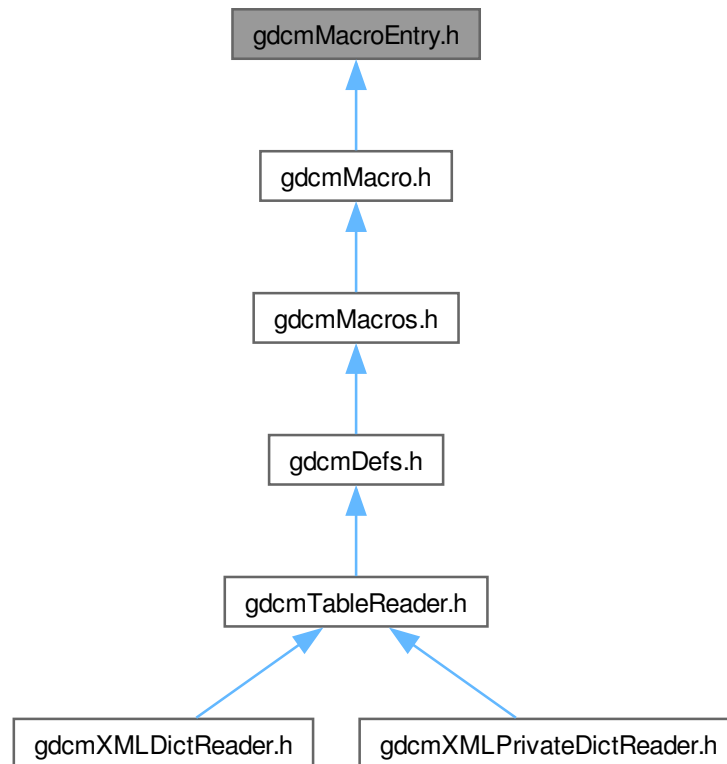
13.213 gdcmMacroEntry.h File Reference

#include "gdcmModuleEntry.h"

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define` [GDCMMACROENTRY_H](#)

13.213.1 Macro Definition Documentation

13.213.1.1 GDCMMACROENTRY_H

`#define` GDCMMACROENTRY_H

13.214 gdcmMacroEntry.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #if 0
00015 #ifndef GDCMMACROENTRY_H
00016 #define GDCMMACROENTRY_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmType.h"
00020
00021 #include <string>
00022
00023 namespace gdcm
00024 {
00025     class GDCM_EXPORT MacroEntry
00026     {
00027     public:
00028         MacroEntry(const char *name = "", const char *type = "3", const char *description =
00029             ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
00030             DataElementType = Type::GetTypeType(type);
00031         }
00032         virtual ~MacroEntry() {} // important
00033         friend std::ostream& operator<<(std::ostream& __os, const MacroEntry &__val);
00034
00035         void SetName(const char *name) { Name = name; }
00036         const char *GetName() const { return Name.c_str(); }
00037
00038         void SetType(const Type &type) { DataElementType = type; }
00039         const Type &GetType() const { return DataElementType; }
00040
00041         /*
00042          * WARNING: 'Description' is currently a std::string, but it might change in the future
00043          * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
00044          * instead.
00045          */
00046         typedef std::string Description;
00047         void SetDescription(const char *d) { DescriptionField = d; }
00048         const Description & GetDescription() const { return DescriptionField; }
00049
00050     protected:
00051         // PS 3.3 repeats the name of an attribute, but often contains typos
00052         // for now we will not use this info, but instead access the DataDict instead
00053         std::string Name;
00054
00055         // An attribute, encoded as a Data Element, may or may not be required in a
00056         // Data Set, depending on that Attribute's Data Element Type.
00057         Type DataElementType;
00058
00059         // TODO: for now contains the raw description (with enumerated values, defined terms...)
00060         Description DescriptionField;
00061     };
00062
00063     //-----
00064     inline std::ostream& operator<<(std::ostream& __os, const MacroEntry &__val)
00065     {
00066         __os << __val.Name << "\t" << __val.DataElementType << "\t" << __val.DescriptionField;
00067         return __os;
00068     }
00069
00070 } // end namespace gdcm
00071
00072 #endif //GDCMMODULEENTRY_H
00073 #endif

```

```

00078
00079 #ifndef GDCMMACROENTRY_H
00080 #define GDCMMACROENTRY_H
00081 #include "gdcmModuleEntry.h"
00082 #endif

```

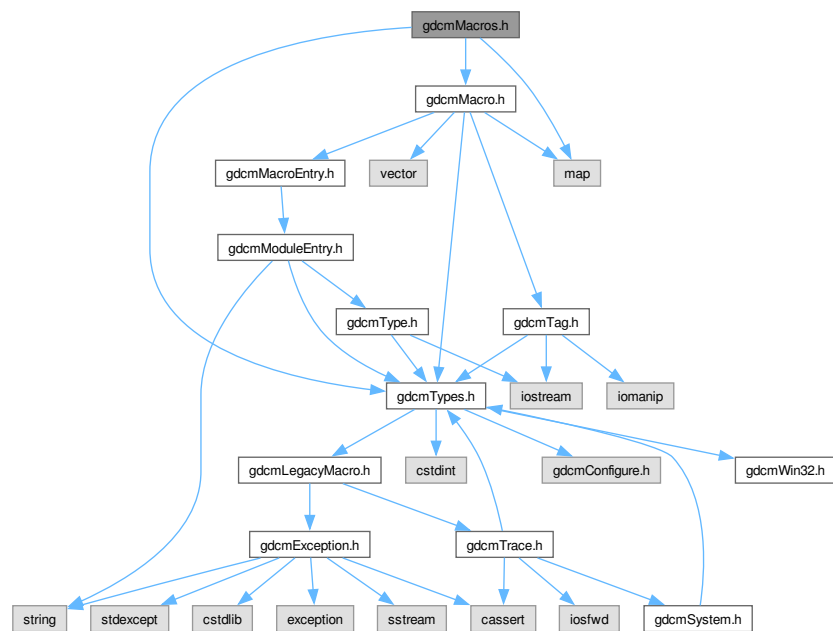
13.215 gdcmMacros.h File Reference

```
#include "gdcmTypes.h"
```

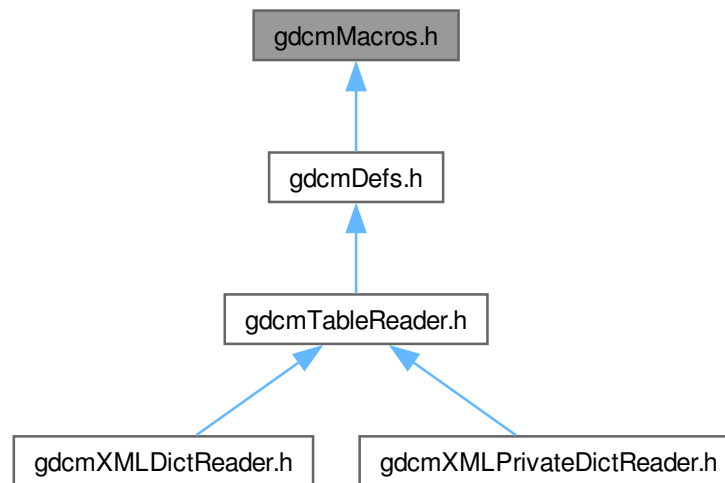
```
#include "gdcmMacro.h"
```

```
#include <map>
```

Include dependency graph for gdcmMacros.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Macros](#)
Class for representing a [Modules](#).

Namespaces

- namespace [gdc](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

13.216 gdcMacros.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even

```



```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00014 #ifndef GDCMMACROS_H
00015 #define GDCMMACROS_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmMacro.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT Macros
00025     {
00026     public:
00027         typedef std::map<std::string, Macro> ModuleMapType;
00028
00029         Macros() = default;
00030         friend std::ostream& operator<<(std::ostream& __os, const Macros& __val);
00031
00032         void Clear() { ModulesInternal.clear(); }
00033
00034         // A Module is inserted based on it's ref
00035         void AddMacro(const char *ref, const Macro & module )
00036         {
00037             gdcm_assert( ref && *ref );
00038             gdcm_assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
00039             ModulesInternal.insert(
00040                 ModuleMapType::value_type(ref, module));
00041         }
00042         const Macro &GetMacro(const char *name) const
00043         {
00044             gdcm_assert( name && *name );
00045             ModuleMapType::const_iterator it = ModulesInternal.find( name );
00046             gdcm_assert( it != ModulesInternal.end() );
00047             gdcm_assert( it->first == name );
00048             return it->second;
00049         }
00050
00051         bool IsEmpty() const { return ModulesInternal.empty(); }
00052
00053     private:
00054         ModuleMapType ModulesInternal;
00055     };
00056
00057 //-----
00058 inline std::ostream& operator<<(std::ostream& __os, const Macros &__val)
00059 {
00060     Macros::ModuleMapType::const_iterator it = __val.ModulesInternal.begin();
00061     for(; it != __val.ModulesInternal.end(); ++it)
00062     {
00063         const std::string &name = it->first;
00064         const Macro &m = it->second;
00065         __os << name << " " << m << "\n";
00066     }
00067     return __os;
00068 }
00069
00070 } // end namespace gdcm
00071
00072 #endif //GDCMMODULES_H

```

13.217 gdcmModule.h File Reference

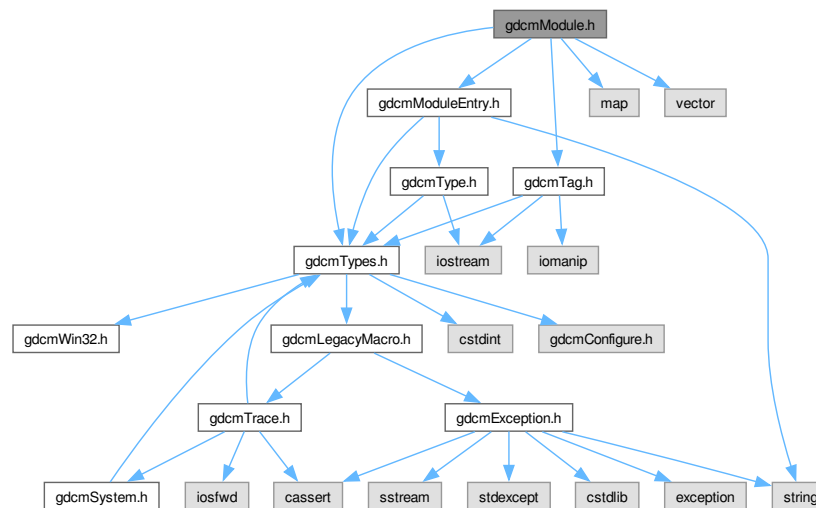
```

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"
#include <map>

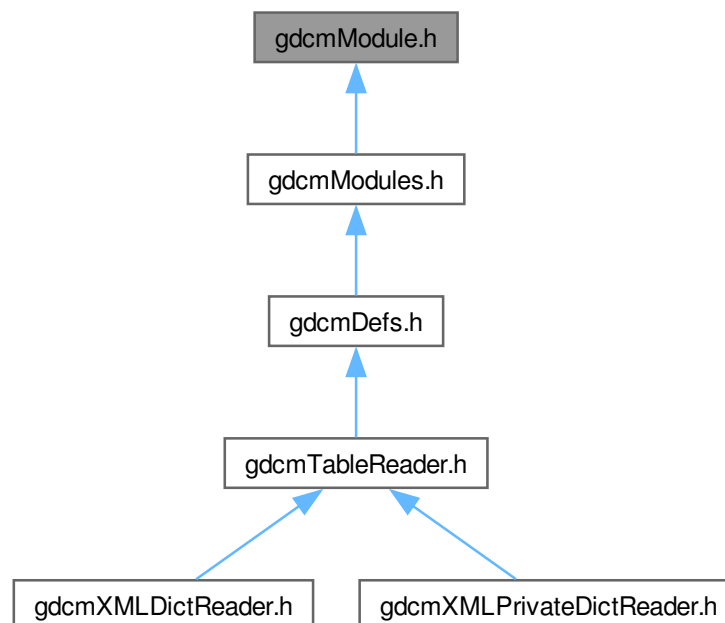
```

```
#include <vector>
```

Include dependency graph for gdcmModule.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Module](#)
Class for representing a [Module](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

13.218 gdcmModule.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMMODULE_H
00015  #define GDCMMODULE_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmTag.h"
00019  #include "gdcmModuleEntry.h"
00020  00021  #include <map>
00022  #include <vector>
00023  00024  namespace gdcm
00025  {
00026  00027  class DataSet;
00028  class Usage;
00029  class Macros;
00037  class GDCM_EXPORT Module
00038  {
00039  public:
00040  typedef std::map<Tag, ModuleEntry> MapModuleEntry;
00041  typedef std::vector<std::string> ArrayIncludeMacrosType;
00042  00043  //typedef MapModuleEntry::const_iterator ConstIterator;
00044  //typedef MapModuleEntry::iterator Iterator;
00045  //ConstIterator Begin() const { return ModuleInternal.begin(); }
00046  //Iterator Begin() { return ModuleInternal.begin(); }
00047  //ConstIterator End() const { return ModuleInternal.end(); }
00048  //Iterator End() { return ModuleInternal.end(); }
00049  00050  Module() = default;
00051  friend std::ostream& operator<<(std::ostream& _os, const Module &_val);
00052  00053  void Clear() { ModuleInternal.clear(); }
00054  00056  void AddModuleEntry(const Tag& tag, const ModuleEntry & module)

```

```

00057     {
00058     ModuleInternal.insert(
00059         MapModuleEntry::value_type(tag, module));
00060     }
00061
00062 void AddMacro(const char *include)
00063 {
00064     ArrayIncludeMacros.push_back( include );
00065 }
00066
00069 bool FindModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
00070 const ModuleEntry& GetModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
00071
00072 void SetName( const char *name) { Name = name; }
00073 const char *GetName() const { return Name.c_str(); }
00074
00075 // Verify will print on std::cerr for error
00076 // Upon success will return true, false otherwise
00077 bool Verify(const DataSet& ds, Usage const & usage) const;
00078
00079 private:
00080 //Module &operator=(const Module &_val); // purposely not implemented
00081 //Module(const Module &_val); // purposely not implemented
00082
00083 MapModuleEntry ModuleInternal;
00084 std::string Name;
00085 ArrayIncludeMacrosType ArrayIncludeMacros;
00086 };
00087 //-----
00088 inline std::ostream& operator<<(std::ostream& __os, const Module &_val)
00089 {
00090     __os << _val.Name << '\n';
00091     Module::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
00092     for(; it != _val.ModuleInternal.end(); ++it)
00093     {
00094         const Tag &t = it->first;
00095         const ModuleEntry &de = it->second;
00096         __os << t << " " << de << '\n';
00097     }
00098
00099     return __os;
00100 }
00101
00102 } // end namespace gdcmm
00103
00104 #endif //GDCMMODULE_H

```

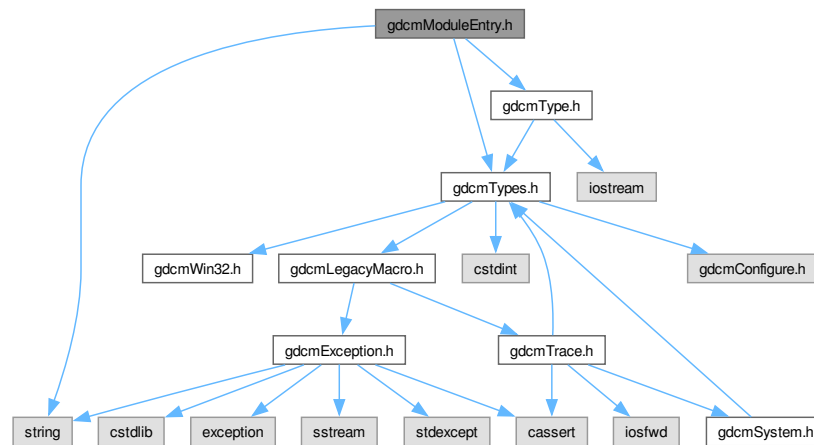
13.219 gdcmmModuleEntry.h File Reference

```

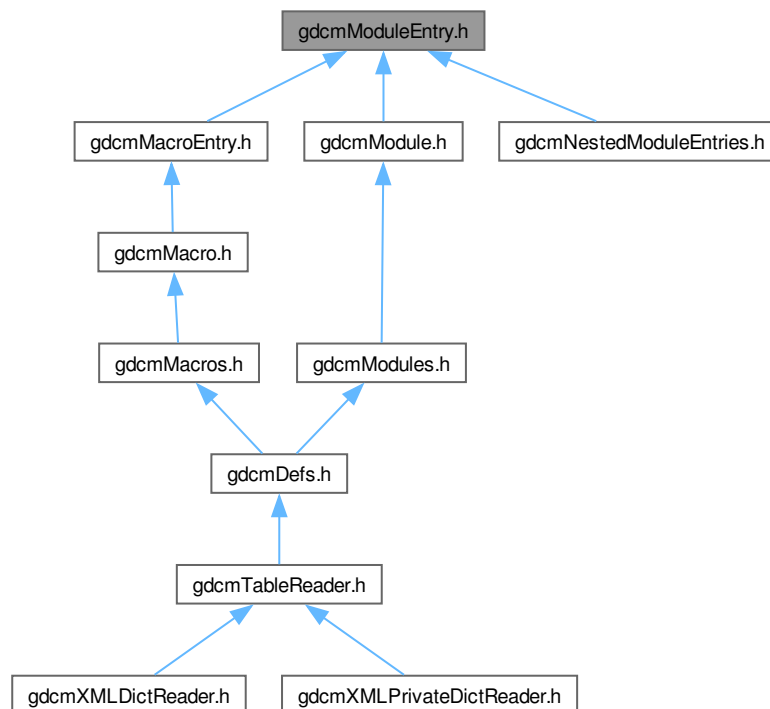
#include "gdcmmTypes.h"
#include "gdcmmType.h"
#include <string>

```

Include dependency graph for gdcmModuleEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ModuleEntry](#)
Class for representing a [ModuleEntry](#).

Namespaces

- namespace [gdcm](#)

Typedefs

- typedef [ModuleEntry](#) [gdcm::MacroEntry](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)`

13.220 [gdcmModuleEntry.h](#)

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMMODULEENTRY_H
00015  #define GDCMMODULEENTRY_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmType.h"
00019
00020  #include <string>
00021
00022  namespace gdcm
00023  {
00024  class GDCM_EXPORT ModuleEntry
00025  {
00026  public:
00027    ModuleEntry(const char *name = "", const char *type = "3", const char *description =
00028    ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
00029      DataElementType = Type::GetTypeType(type);
00030    }
00031    virtual ~ModuleEntry() = default; // important
00032    friend std::ostream& operator<<(std::ostream& _os, const ModuleEntry &_val);
00033
00034    void SetName(const char *name) { Name = name; }
00035    const char *GetName() const { return Name.c_str(); }
00036
00037    void SetType(const Type &type) { DataElementType = type; }
00038    const Type &GetType() const { return DataElementType; }
00039
00040  /*

```

```

00045  * WARNING: 'Description' is currently a std::string, but it might change in the future
00046  * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
00047  * instead.
00048  */
00049  typedef std::string Description;
00050  void SetDescription(const char *d) { DescriptionField = d; }
00051  const Description & GetDescription() const { return DescriptionField; }
00052
00053 protected:
00054  // PS 3.3 repeats the name of an attribute, but often contains typos
00055  // for now we will not use this info, but instead access the DataDict instead
00056  std::string Name;
00057
00058  // An attribute, encoded as a Data Element, may or may not be required in a
00059  // Data Set, depending on that Attribute's Data Element Type.
00060  Type DataElementType;
00061
00062  // TODO: for now contains the raw description (with enumerated values, defined terms...)
00063  Description DescriptionField;
00064 };
00065 //-----
00066 inline std::ostream& operator<<(std::ostream& _os, const ModuleEntry &_val)
00067 {
00068   _os << _val.Name << "\\t" << _val.DataElementType << "\\t" << _val.DescriptionField;
00069   return _os;
00070 }
00071
00072 typedef ModuleEntry MacroEntry;
00073
00074
00075 } // end namespace gdc
00076
00077 #endif //GDCMMODULEENTRY_H

```

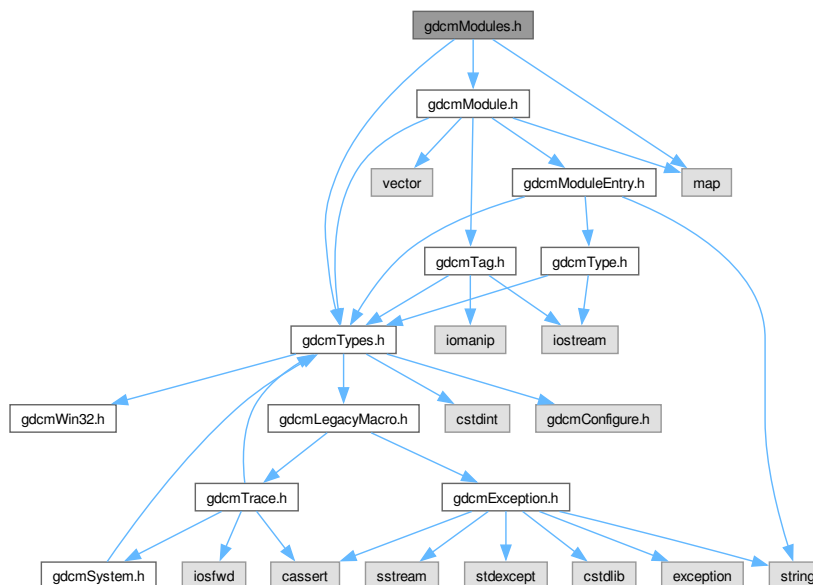
13.221 gdcModules.h File Reference

```

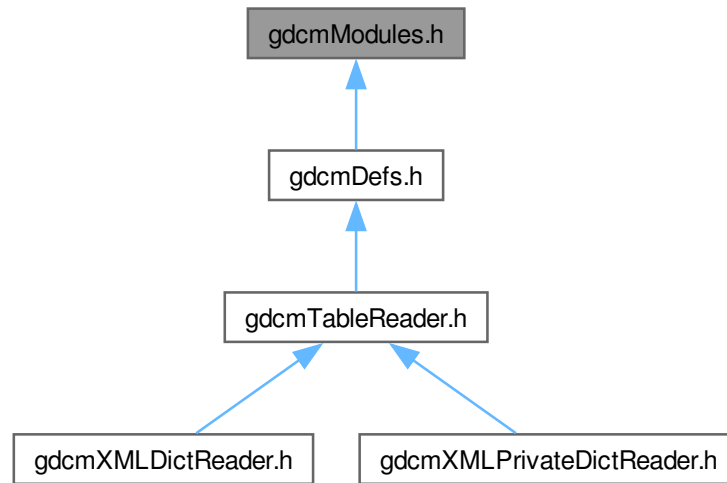
#include "gdcTypes.h"
#include "gdcModule.h"
#include <map>

```

Include dependency graph for gdcModules.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Modules](#)
Class for representing a [Modules](#).

Namespaces

- namespace [gdc](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

13.222 gdcModules.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even

```



```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00014 #ifndef GDCMMODULES_H
00015 #define GDCMMODULES_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmModule.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024     class GDCM_EXPORT Modules
00025     {
00026     public:
00027         typedef std::map<std::string, Module> ModuleMapType;
00028
00029         Modules() = default;
00030         friend std::ostream& operator<<(std::ostream& __os, const Modules &__val);
00031
00032         void Clear() { ModulesInternal.clear(); }
00033
00034         // A Module is inserted based on it's ref
00035         void AddModule(const char *ref, const Module & module )
00036         {
00037             gdcm_assert( ref && *ref );
00038             gdcm_assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
00039             ModulesInternal.insert(
00040                 ModuleMapType::value_type(ref, module));
00041         }
00042         const Module &GetModule(const char *name) const
00043         {
00044             gdcm_assert( name && *name );
00045             ModuleMapType::const_iterator it = ModulesInternal.find( name );
00046             gdcm_assert( it != ModulesInternal.end() );
00047             gdcm_assert( it->first == name );
00048             return it->second;
00049         }
00050
00051         bool IsEmpty() const { return ModulesInternal.empty(); }
00052
00053     private:
00054         ModuleMapType ModulesInternal;
00055     };
00056
00057 //-----
00058 inline std::ostream& operator<<(std::ostream& __os, const Modules &__val)
00059 {
00060     Modules::ModuleMapType::const_iterator it = __val.ModulesInternal.begin();
00061     for(;it != __val.ModulesInternal.end(); ++it)
00062     {
00063         const std::string &name = it->first;
00064         const Module &m = it->second;
00065         __os << name << " " << m << "\n";
00066     }
00067
00068     return __os;
00069 }
00070
00071 } // end namespace gdcm
00072
00073 #endif //GDCMMODULES_H

```

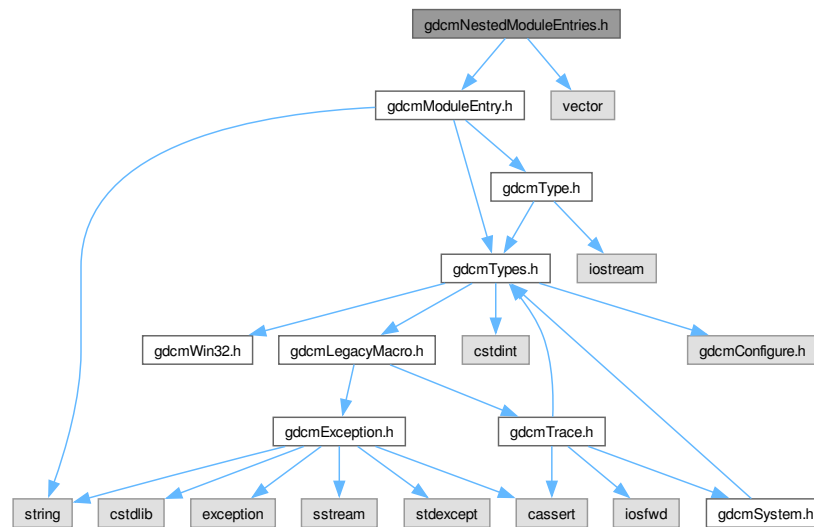
13.223 gdcmNestedModuleEntries.h File Reference

```

#include "gdcmModuleEntry.h"
#include <vector>

```

Include dependency graph for `gdcmNestedModuleEntries.h`:



Classes

- class `gdcm::NestedModuleEntries`
Class for representing a `NestedModuleEntries`.

Namespaces

- namespace `gdcm`

Typedefs

- typedef `NestedModuleEntries` `gdcm::NestedMacroEntries`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

13.224 gdcnNestedModuleEntries.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcn.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMNESTEDMODULEENTRIES_H
00015 #define GDCMNESTEDMODULEENTRIES_H
00016
00017 #include "gdcnModuleEntry.h"
00018 #include <vector>
00019
00020 namespace gdcn
00021 {
00022     class GDCM_EXPORT NestedModuleEntries : public ModuleEntry
00023     {
00024     public:
00025         NestedModuleEntries(const char *name = "", const char *type = "3", const char *description =
00026             ""):ModuleEntry(name,type,description) { }
00027         friend std::ostream& operator<(std::ostream& __os, const NestedModuleEntries &__val);
00028
00029         typedef std::vector<ModuleEntry>::size_type SizeType;
00030         SizeType GetNumberOfModuleEntries() { return ModuleEntriesList.size(); }
00031
00032         const ModuleEntry &GetModuleEntry(SizeType idx) const { return ModuleEntriesList[idx]; }
00033         ModuleEntry &GetModuleEntry(SizeType idx) { return ModuleEntriesList[idx]; }
00034
00035         void AddModuleEntry(const ModuleEntry &me) { ModuleEntriesList.push_back( me ); }
00036
00037     private:
00038         std::vector<ModuleEntry> ModuleEntriesList;
00039     };
00040
00041 //-----
00042 inline std::ostream& operator<(std::ostream& __os, const NestedModuleEntries &__val)
00043 {
00044     __os << "Nested:" << __val.Name << "\t" << __val.DataElementType << "\t" << __val.DescriptionField;
00045     return __os;
00046 }
00047
00048 typedef NestedModuleEntries NestedMacroEntries;
00049
00050 } // end namespace gdcn
00051
00052 #endif //GDCMNESTEDMODULEENTRIES_H

```

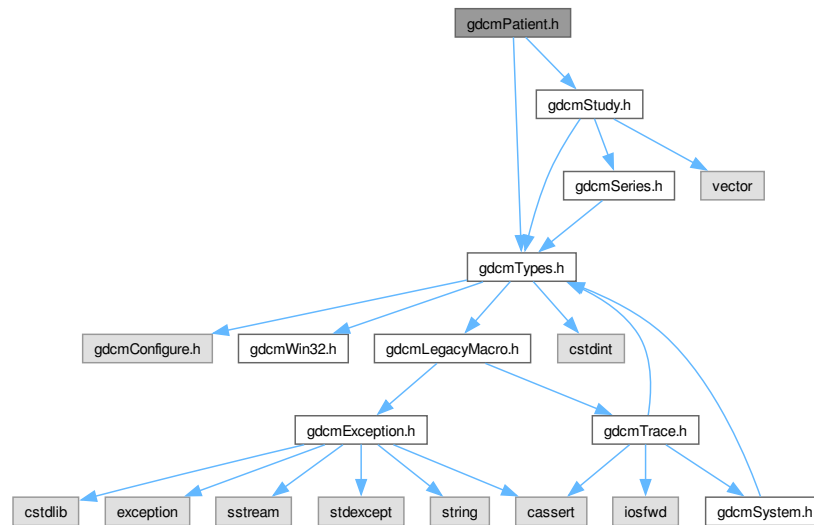
13.225 gdcnPatient.h File Reference

```

#include "gdcnTypes.h"
#include "gdcnStudy.h"

```

Include dependency graph for `gdcmPatient.h`:



Classes

- class [gdcm::Patient](#)

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

Namespaces

- namespace [gdcm](#)

13.226 `gdcmPatient.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMPATIENT_H
00015  #define GDCMPATIENT_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmStudy.h"

```

```

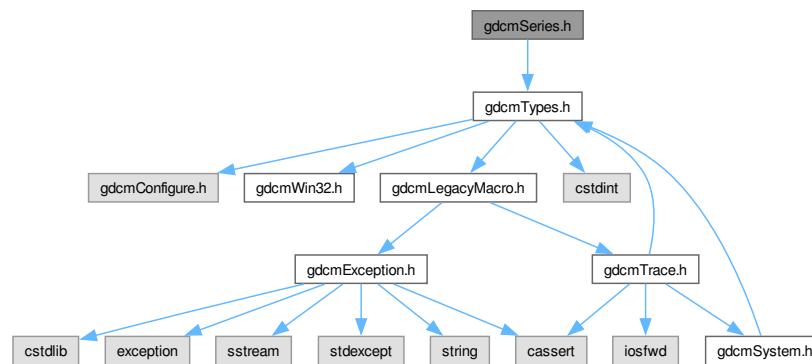
00019
00020 namespace gdc
00021 {
00022 class GDCM_EXPORT Patient
00023 {
00024 public:
00025 Patient() = default;
00026
00027 private:
00028 std::vector<Study> StudyList;
00029 };
00030
00031 } // end namespace gdc
00032
00033 #endif //GDCMPATIENT_H

```

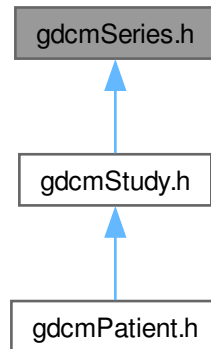
13.227 gdcSeries.h File Reference

#include "gdcTypes.h"

Include dependency graph for gdcSeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcms::Series](#)
[Series](#).

Namespaces

- namespace [gdcms](#)

13.228 gdcmsSeries.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMSERIES_H
00015  #define GDCMSERIES_H
00016  00017  #include "gdcmsTypes.h"
00018  00019  namespace gdcms
00020  {
00024  class GDCM_EXPORT Series
  
```

```

00025 {
00026 public:
00027     Series() = default;
00028 private:
00029     // Image, Waveform...
00030 };
00031
00032 } // end namespace gdcm
00033
00034 #endif //GDCMSERIES_H

```

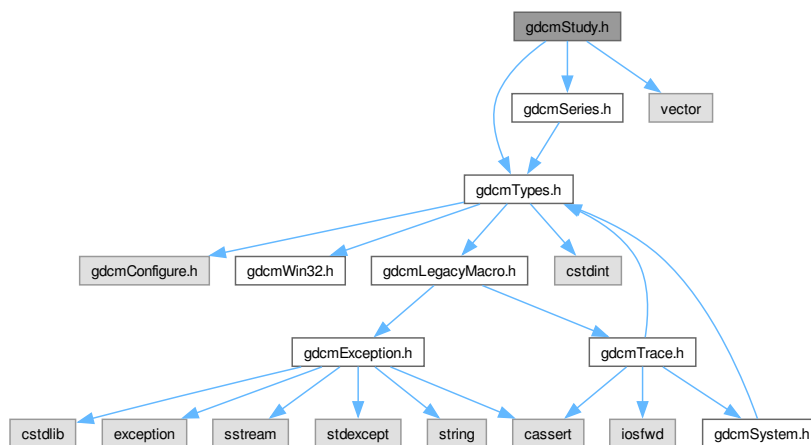
13.229 gdcmStudy.h File Reference

```
#include "gdcmTypes.h"
```

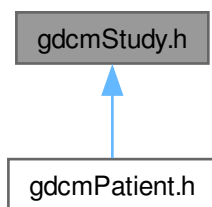
```
#include "gdcmSeries.h"
```

```
#include <vector>
```

Include dependency graph for gdcmStudy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Study](#)
[Study](#).

Namespaces

- namespace [gdcm](#)

13.230 [gdcmStudy.h](#)

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSTUDY_H
00015  #define GDCMSTUDY_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmSeries.h"
00019
00020  #include <vector>
00021
00022  namespace gdcm
00023  {
00024  00027  class GDCM_EXPORT Study
00028  {
00029  public:
00030  Study() = default;
00031  private:
00032  std::vector<Series> SeriesList;
00033  };
00034
00035  } // end namespace gdcm
00036
00037  #endif //GDCMSTUDY_H

```

13.231 [gdcmTable.h](#) File Reference

```

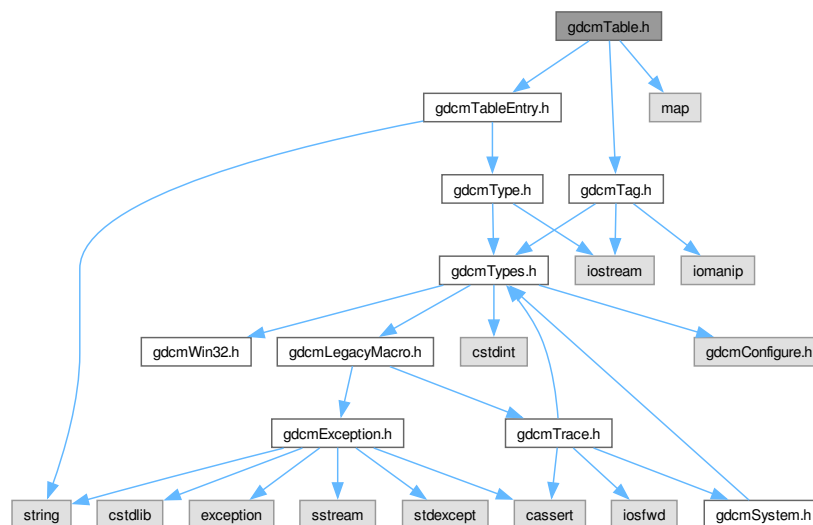
#include "gdcmTableEntry.h"
#include "gdcmTag.h"

```



```
#include <map>
```

Include dependency graph for gdcTable.h:



Classes

- class `gdc::Table`
`Table`.

Namespaces

- namespace `gdc`

13.232 gdcTable.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMTABLE_H
00015 #define GDCMTABLE_H
00016

```

```

00017 #include "gdcmTableEntry.h"
00018 #include "gdcmTag.h"
00019
00020 #include <map>
00021
00022 namespace gdcm
00023 {
00024
00028 class Table
00029 {
00030 public:
00031     typedef std::map<Tag, TableEntry> MapTableEntry;
00032     Table() = default;
00033     ~Table() = default;
00034     Table &operator=(const Table &_val) = delete;
00035     Table(const Table&_val) = delete;
00036
00037     friend std::ostream& operator<<(std::ostream& __os, const Table &_val);
00038
00039     void InsertEntry(Tag const &tag, TableEntry const &te)
00040     {
00041         #ifndef NDEBUB
00042             MapTableEntry::size_type s = TableInternal.size();
00043         #endif
00044         TableInternal.insert(
00045             MapTableEntry::value_type(tag, te));
00046         #ifndef NDEBUB
00047             gdcm_assert( s < TableInternal.size() );
00048         #endif
00049     }
00050
00051     const TableEntry &GetTableEntry(const Tag &tag) const
00052     {
00053         MapTableEntry::const_iterator it =
00054             TableInternal.find(tag);
00055         if (it == TableInternal.end())
00056         {
00057             gdcm_assert( 0 && "Impossible" );
00058             return GetTableEntry(Tag(0,0));
00059         }
00060         return it->second;
00061     }
00062
00063     MapTableEntry TableInternal;
00064 };
00065
00066 } // end namespace gdcm
00067
00068 #endif //GDCMTABLE_H

```

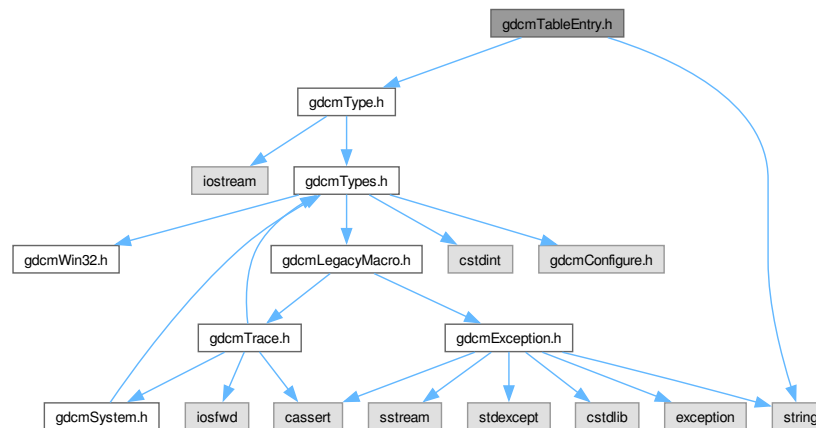
13.233 gdcmTableEntry.h File Reference

```

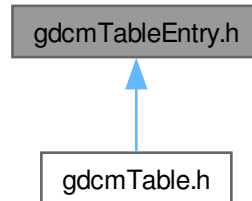
#include "gdcmType.h"
#include <string>

```

Include dependency graph for gdcTableEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::TableEntry`
`TableEntry`.

Namespaces

- namespace `gdc`

13.234 gdcmTableEntry.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTABLEENTRY_H
00015 #define GDCMTABLEENTRY_H
00016
00017 #include "gdcmType.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023
00027 class TableEntry
00028 {
00029 public:
00030     TableEntry(const char *attribute = nullptr,
00031               Type const &type = Type(), const char *des = nullptr) :
00032         Attribute(attribute ? attribute : ""), TypeField(type), Description(des ? des : "") {}
00033     ~TableEntry() = default;
00034
00035 private:
00036     std::string Attribute;
00037     Type TypeField;
00038     std::string Description;
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMTABLEENTRY_H

```

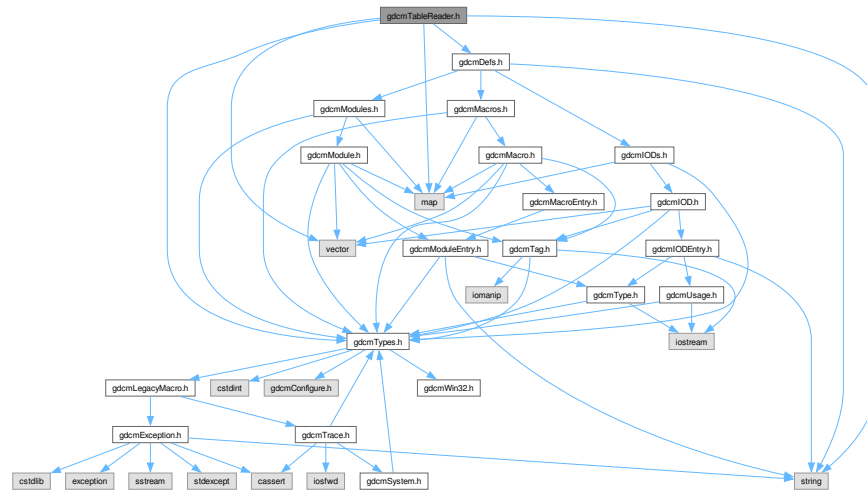
13.235 gdcmTableReader.h File Reference

```

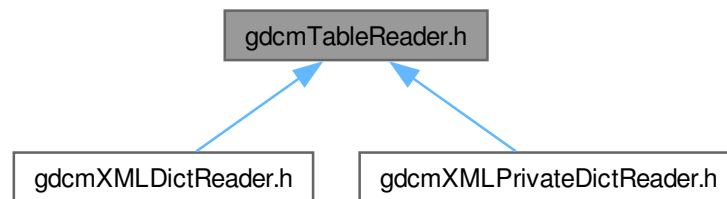
#include "gdcmTypes.h"
#include "gdcmDefs.h"
#include <string>
#include <vector>
#include <map>

```

Include dependency graph for gdcmTableReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::TableReader`
Class for representing a `TableReader`.

Namespaces

- namespace `gdcm`

13.236 gdcmTableReader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTABLEREADER_H
00015 #define GDCMTABLEREADER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmDefs.h"
00019 // #include "gdcmModule.h"
00020 // #include "gdcmIOD.h"
00021 // #include "gdcmIODs.h"
00022 // #include "gdcmModules.h"
00023
00024 #include <string>
00025 #include <vector>
00026 #include <map>
00027
00028 namespace gdcm
00029 {
00030     class GDCM_EXPORT TableReader
00031     {
00032     public:
00033         TableReader(Defs &defs):CurrentDefs(defs),ParsingModule(false),ParsingModuleEntry(false),
00034             ParsingModuleEntryDescription(false),
00035             ParsingMacro(false),
00036             ParsingMacroEntry(false),
00037             ParsingMacroEntryDescription(false),
00038             ParsingIOD(false),
00039             ParsingIODEntry(false),
00040             Description() {}
00041         virtual ~TableReader() = default;
00042
00043         // Set/Get filename
00044         void SetFilename(const char *filename) { Filename = filename; }
00045         const char *GetFilename() { return Filename.c_str(); }
00046
00047         int Read();
00048
00049     protected:
00050         // You need to override those function in your subclasses:
00051         virtual void StartElement(const char *name, const char **atts);
00052         virtual void EndElement(const char *name);
00053         virtual void CharacterDataHandler(const char *data, int length);
00054
00055         void HandleModuleEntry(const char **atts);
00056         void HandleModule(const char **atts);
00057         void HandleModuleEntryDescription(const char **atts);
00058         void HandleMacroEntry(const char **atts);
00059         void HandleMacro(const char **atts);
00060         void HandleMacroEntryDescription(const char **atts);
00061         void HandleModuleInclude(const char **atts);
00062         void HandleIODEntry(const char **atts);
00063         void HandleIOD(const char **atts);
00064
00065         //const Modules & GetModules() const { return CurrentModules; }
00066         //const Macros & GetMacros() const { return CurrentMacros; }
00067         //const IODs & GetIODs() const { return CurrentIODs; }
00068         const Defs & GetDefs() const { return CurrentDefs; }
00069
00070     private:
00071         std::string Filename;
00072         Defs &CurrentDefs;
00073         //Macros CurrentMacros;

```

```

00078 //Modules CurrentModules;
00079 //IODs CurrentIODs;
00080 Macro CurrentMacro;
00081 Module CurrentModule;
00082 IOD CurrentIOD;
00083 MacroEntry CurrentMacroEntry;
00084 ModuleEntry CurrentModuleEntry;
00085 IODEntry CurrentIODEntry;
00086 std::string CurrentModuleName;
00087 std::string CurrentModuleRef;
00088 std::string CurrentMacroRef;
00089 bool ParsingModule;
00090 bool ParsingModuleEntry;
00091 bool ParsingModuleEntryDescription;
00092 bool ParsingMacro;
00093 bool ParsingMacroEntry;
00094 bool ParsingMacroEntryDescription;
00095 bool ParsingIOD;
00096 bool ParsingIODEntry;
00097 Tag CurrentTag;
00098 std::string Description;
00099 };
00100
00101 } // end namespace gdcm
00102
00103 #endif //GDCMTABLEREADER_H

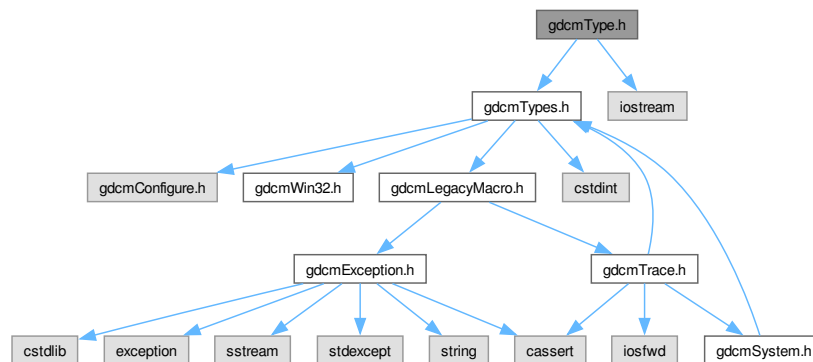
```

13.237 gdcmType.h File Reference

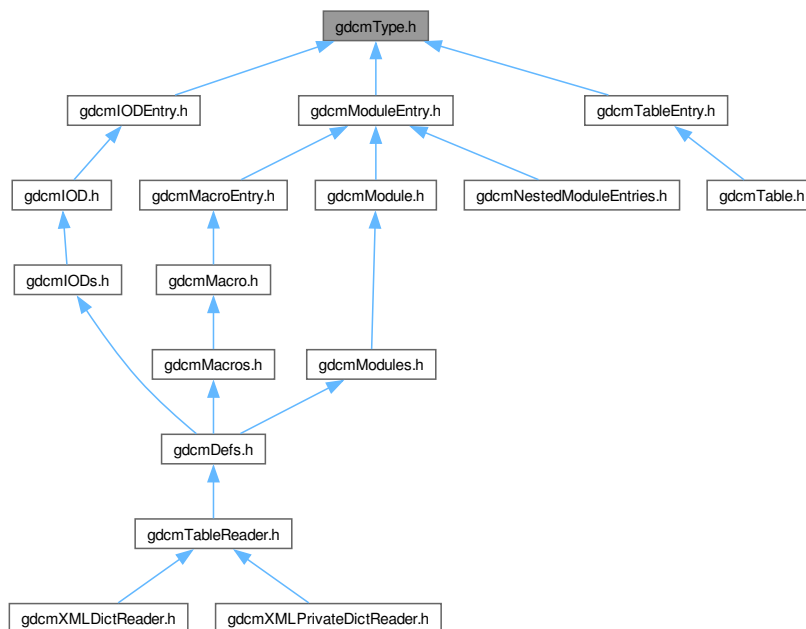
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmType.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Type](#)
[Type](#).

Namespaces

- namespace [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

13.238 gdcmType.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  Copyright (c) 2006-2011 Mathieu Malaterre
00005  All rights reserved.

```



```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00015 #ifndef GDCMTYPE_H
00016 #define GDCMTYPE_H
00017
00018 #include "gdcmTypes.h"
00019
00020 #include <iostream>
00021
00022 namespace gdcm
00023 {
00024
00041 class GDCM_EXPORT Type
00042 {
00043 public:
00044     typedef enum {
00045         T1 = 0,
00046         T1C,
00047         T2,
00048         T2C,
00049         T3,
00050         UNKNOWN
00051     } TypeType;
00052
00053     Type(TypeType type = UNKNOWN) : TypeField(type) { }
00054
00055     operator TypeType () const { return TypeField; }
00056     friend std::ostream &operator<<(std::ostream &os, const Type &vr);
00057
00058     static const char *GetTypeString(TypeType type);
00059     static TypeType GetTypeType(const char *type);
00060
00061 private:
00062     TypeType TypeField;
00063 };
00064 //-----
00065 inline std::ostream &operator<<(std::ostream &_os, const Type &val)
00066 {
00067     _os << Type::GetTypeString(val.TypeField);
00068     return _os;
00069 }
00070
00071 } // end namespace gdcm
00072
00073 #endif //GDCMTYPE_H

```

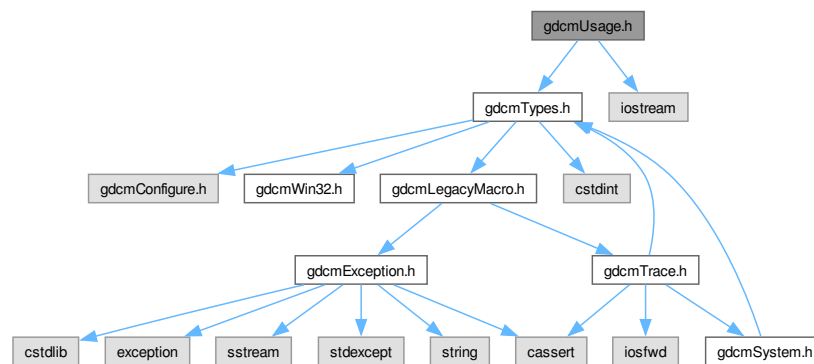
13.239 gdcmUsage.h File Reference

```

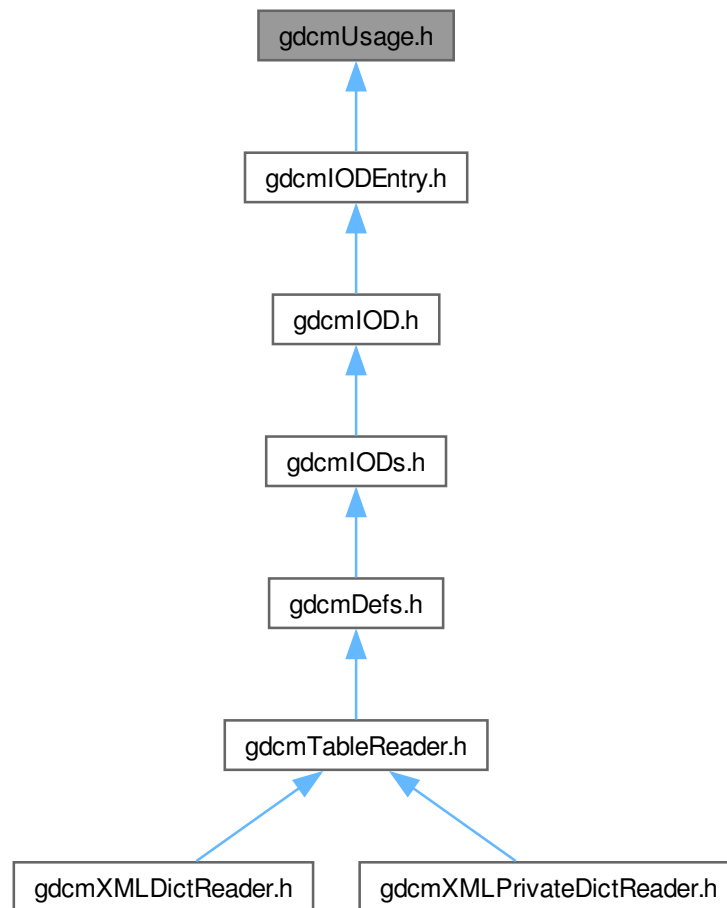
#include "gdcmTypes.h"
#include <iostream>

```

Include dependency graph for `gdcmlUsage.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::Usage](#)
[Usage](#).

Namespaces

- namespace [gdcml](#)

Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const Usage &val)`

13.240 gdcmUsage.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMUSAGE_H
00015 #define GDCMUSAGE_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <iostream>
00020
00021 namespace gdcm
00022 {
00023
00048 class GDCM__EXPORT Usage
00049 {
00050 public:
00051     typedef enum {
00052         Mandatory, // (see A.1.3.1) , abbreviated M
00053         Conditional, // (see A.1.3.2) , abbreviated C
00054         UserOption, // (see A.1.3.3) , abbreviated U
00055         Invalid
00056     } UsageType;
00057
00058     Usage(UsageType type = Invalid) : UsageField(type) { }
00059
00060     operator UsageType () const { return UsageField; }
00061     friend std::ostream &operator<<(std::ostream &os, const Usage &vr);
00062
00063     static const char *GetUsageString(UsageType type);
00064     static UsageType GetUsageType(const char *type);
00065
00066 private:
00067     UsageType UsageField;
00068 };
00069 //-----
00070 inline std::ostream &operator<<(std::ostream &_os, const Usage &val)
00071 {
00072     _os << Usage::GetUsageString(val.UsageField);
00073     return _os;
00074 }
00075
00076 } // end namespace gdcm
00077
00078 #endif //GDCMUSAGE_H

```

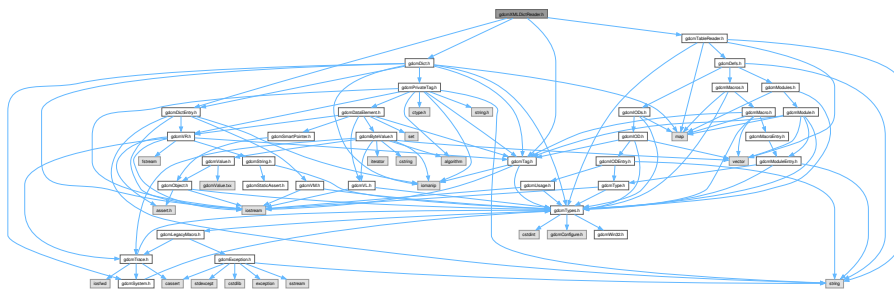
13.241 gdcmXMLDictReader.h File Reference

```

#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"

```

Include dependency graph for gdcmXMLDictReader.h:



Classes

- class [gdcm::XMLDictReader](#)
Class for representing a [XMLDictReader](#).

Namespaces

- namespace [gdcm](#)

13.242 gdcmXMLDictReader.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMXMLDICTREADER_H
00015  #define GDCMXMLDICTREADER_H
00016
00017  #include "gdcmTableReader.h"
00018  #include "gdcmDict.h"
00019  #include "gdcmDictEntry.h"
00020  #include "gdcmTag.h"
00021
00022  namespace gdcm
00023  {
00024  {
00029  class GDCM_EXPORT XMLDictReader : public TableReader
00030  {
00031  public:
00032    XMLDictReader();
00033    ~XMLDictReader() {}
00034
00035    void StartElement(const char *name, const char **atts);
00036    void EndElement(const char *name);
00037    void CharacterDataHandler(const char *data, int length);
00038  }

```


13.244 gdcmlXMLPrivateDictReader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMLXMLPRIVATEDICTREADER_H
00015 #define GDCMLXMLPRIVATEDICTREADER_H
00016
00017 #include "gdcmlTableReader.h"
00018 #include "gdcmlDict.h"
00019 #include "gdcmlDictEntry.h"
00020 #include "gdcmlTag.h"
00021
00022 namespace gdcml
00023 {
00024     class GDCML_EXPORT XMLPrivateDictReader : public TableReader
00025     {
00026     public:
00027         XMLPrivateDictReader();
00028         ~XMLPrivateDictReader() {}
00029
00030         void StartElement(const char *name, const char **atts);
00031         void EndElement(const char *name);
00032         void CharacterDataHandler(const char *data, int length);
00033
00034         const PrivateDict & GetPrivateDict() { return PDict; }
00035
00036     protected:
00037         void HandleEntry(const char **atts);
00038         void HandleDescription(const char **atts);
00039
00040     private:
00041         PrivateDict PDict;
00042         PrivateTag CurrentTag;
00043         DictEntry CurrentDE;
00044         bool ParsingDescription;
00045         std::string Description;
00046     };
00047 } // end namespace gdcml
00048
00049 #endif //GDCMLXMLPRIVATEDICTREADER_H

```

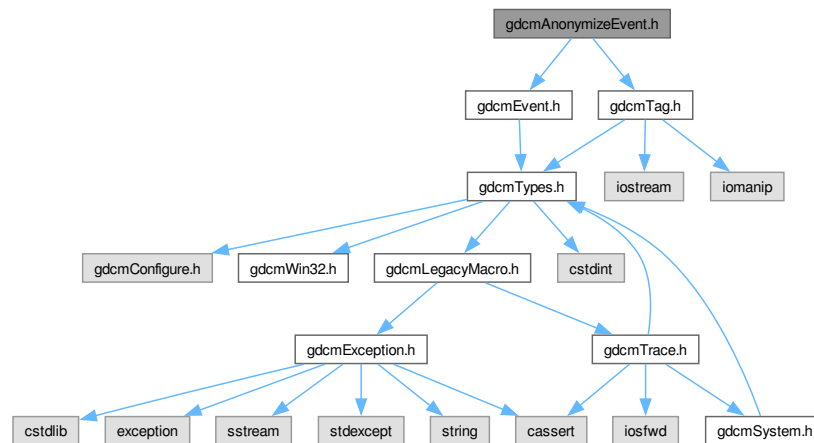
13.245 gdcmlAnonymizeEvent.h File Reference

```

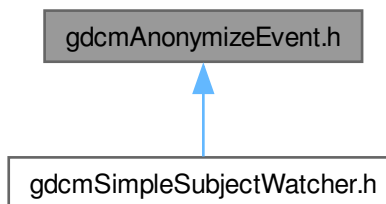
#include "gdcmlEvent.h"
#include "gdcmlTag.h"

```

Include dependency graph for `gdcmAnonymizeEvent.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::AnonymizeEvent`
`AnonymizeEvent`.

Namespaces

- namespace `gdcm`

13.246 gdcmAnonymizeEvent.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMANONYMIZEEVENT_H
00015  #define GDCMANONYMIZEEVENT_H
00016  00017  #include "gdcmEvent.h"
00018  #include "gdcmTag.h"
00019  00020  namespace gdcm
00021  {
00022  00023  class AnonymizeEvent : public AnyEvent
00024  {
00025  public:
00026  typedef AnonymizeEvent Self;
00027  typedef AnyEvent Superclass;
00028  AnonymizeEvent(Tag const &tag = 0):m_Tag(tag) {}
00029  ~AnonymizeEvent() override = default;
00030  AnonymizeEvent(const Self&s) : AnyEvent(s){}
00031  void operator=(const Self&) = delete;
00032  00033  const char * GetEventName() const override { return "AnonymizeEvent"; }
00034  bool CheckEvent(const ::gdcm::Event* e) const override
00035  { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
00036  ::gdcm::Event* MakeObject() const override
00037  { return new Self; }
00038  00039  void SetTag(const Tag& t) { m_Tag = t; }
00040  Tag const & GetTag() const { return m_Tag; }
00041  private:
00042  Tag m_Tag;
00043  };
00044  00045  } // end namespace gdcm
00046  00047  #endif //GDCMANONYMIZEEVENT_H

```

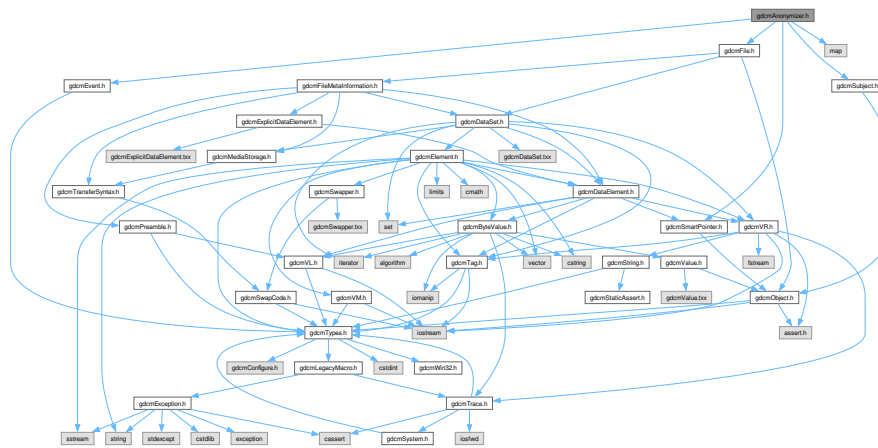
13.247 gdcmAnonymizer.h File Reference

```

#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>

```

Include dependency graph for `gdcmAnonymizer.h`:



Classes

- class `gdcm::Anonymizer`
`Anonymizer`.

Namespaces

- namespace `gdcm`

13.248 gdcmAnonymizer.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMANONYMIZER_H
00015  #define GDCMANONYMIZER_H
00016
00017  #include "gdcmFile.h"
00018  #include "gdcmSubject.h"
00019  #include "gdcmEvent.h"
00020  #include "gdcmSmartPointer.h"
00021
00022  #include <map>
00023
00024  namespace gdcm
00025  {

```

```

00026 class TagPath;
00027 class IOD;
00028 class CryptographicMessageSyntax;
00029
00077 class GDCM_EXPORT Anonymizer : public Subject
00078 {
00079 public:
00080     Anonymizer():F(new File),CMS(nullptr) {}
00081     ~Anonymizer() override;
00082
00084     bool Empty( Tag const &t );
00085
00090     bool Empty( PrivateTag const &pt );
00091
00093     bool Clear( Tag const &t );
00094     bool Clear( PrivateTag const &pt );
00095
00097     bool Remove( Tag const &t );
00098
00104     bool Remove( PrivateTag const &pt );
00105
00108     bool Replace( Tag const &t, const char *value );
00109     bool Replace( PrivateTag const &t, const char *value );
00110
00113     bool Replace( Tag const &t, const char *value, VL const & vl );
00114     bool Replace( PrivateTag const &t, const char *value, VL const & vl );
00115
00117     bool RemovePrivateTags();
00118
00120     bool RemoveGroupLength();
00121
00123     bool RemoveRetired();
00124
00126     void SetFile(const File& f) { F = f; }
00127     //const File &GetFile() const { return *F; }
00128     File &GetFile() { return *F; }
00129
00134     bool BasicApplicationLevelConfidentialityProfile(bool deidentify = true);
00135
00137     void SetCryptographicMessageSyntax( CryptographicMessageSyntax *cms );
00138     const CryptographicMessageSyntax *GetCryptographicMessageSyntax() const;
00139
00141     static SmartPointer<Anonymizer> New() { return new Anonymizer; }
00142
00144     static std::vector<Tag> GetBasicApplicationLevelConfidentialityProfileAttributes();
00145
00148     static void ClearInternalUIDs();
00149
00150 protected:
00151     // Internal function used to either empty a tag or set it's value to a dummy value (Type 1 vs Type 2)
00152     bool BALCPPProtect(DataSet &ds, Tag const & tag, const IOD &ioid);
00153     bool CanEmptyTag(Tag const &tag, const IOD &ioid) const;
00154     void RecurseDataSet( DataSet & ds );
00155
00156 private:
00157     bool BasicApplicationLevelConfidentialityProfile1();
00158     bool BasicApplicationLevelConfidentialityProfile2();
00159     bool CheckIfSequenceContainsAttributeToAnonymize(File const &file, SequenceOfItems* sqi) const;
00160
00161 private:
00162     // I would prefer to have a smart pointer to DataSet but DataSet does not derive from Object...
00163     SmartPointer<File> F;
00164     CryptographicMessageSyntax *CMS;
00165
00166     typedef std::pair< Tag, std::string > TagValueKey;
00167     typedef std::map< TagValueKey, std::string > DummyMapNonUIDTags;
00168     typedef std::map< std::string, std::string > DummyMapUIDTags;
00169     static DummyMapNonUIDTags dummyMapNonUIDTags;
00170     static DummyMapUIDTags dummyMapUIDTags;
00171 };
00172
00178
00179 } // end namespace gdcmanonymizer
00180
00181 #endif //GDCMANONYMIZER_H

```

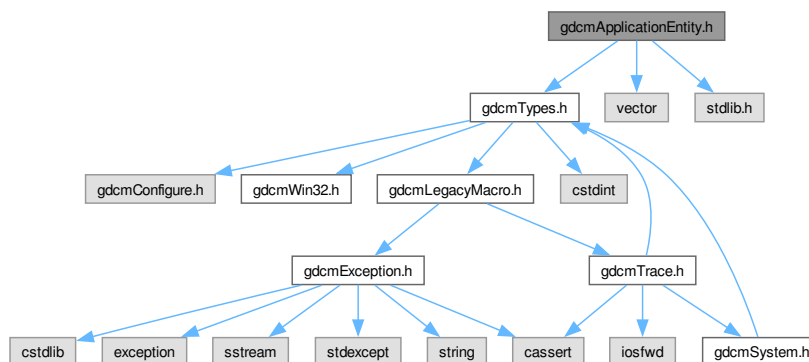
13.249 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <vector>
```

```
#include <stdlib.h>
```

Include dependency graph for gdcmApplicationEntity.h:



Classes

- class [gdcm::ApplicationEntity](#)
[ApplicationEntity](#).

Namespaces

- namespace [gdcm](#)

13.250 gdcmApplicationEntity.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMAPPLICATIONENTITY_H
00015  #define GDCMAPPLICATIONENTITY_H
00016
00017  #include "gdcmTypes.h"

```

```

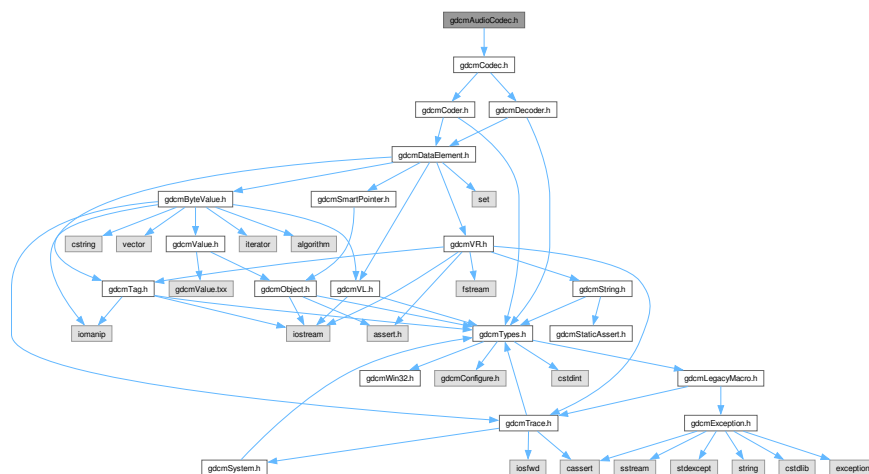
00018 #include <vector>
00019 #include <stdlib.h> // abort
00020
00021 namespace gdcm
00022 {
00023
00035 class GDCM_EXPORT ApplicationEntity
00036 {
00037 public:
00038     static const unsigned int MaxNumberOfComponents = 1;
00039     static const unsigned int MaxLength = 16;
00040     std::string Internal;
00041     static const char Separator = ' ';
00042     static const char Padding = ' ';
00043     //static const char Excluded[5] = { '\\', /* 5CH */, '\n', /* LF */, '\f', /* FF */, '\r', /* CR */, 0x1b /* ESC */ };
00044
00045     bool IsValid() const {
00046         return true;
00047     }
00048     void Squeeze() {
00049         // trim leading and trailing white spaces
00050     }
00051     void SetBlob(const std::vector<char>& v) {
00052         (void)v;
00053         gdcm_assert(0); //TODO
00054     }
00055     void Print(std::ostream &os) const {
00056         (void)os;
00057         gdcm_assert(0); //TODO
00058     }
00059 };
00060
00061 } // end namespace gdcm
00062
00063 #endif //GDCMAPPLICATIONENTITY_H

```

13.251 gdcmAudioCodec.h File Reference

#include "gdcmCodec.h"

Include dependency graph for gdcmAudioCodec.h:



Classes

- class `gdcm::AudioCodec`
`AudioCodec`.

Namespaces

- namespace [gdcm](#)

13.252 gdcmAudioCodec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMAUDIOCODEC_H
00015  #define GDCMAUDIOCODEC_H
00016  00017  #include "gdcmCodec.h"
00018  00019  namespace gdcm
00020  {
00021  00025  class GDCM_EXPORT AudioCodec : public Codec
00026  {
00027  public:
00028  AudioCodec();
00029  ~AudioCodec() override;
00030  bool CanCode(TransferSyntax const &) const override { return false; }
00031  bool CanDecode(TransferSyntax const &) const override { return false; }
00032  bool Decode(DataElement const &is, DataElement &os) override;
00033  };
00034  00035  } // end namespace gdcm
00036  00037  #endif //GDCMAUDIOCODEC_H

```

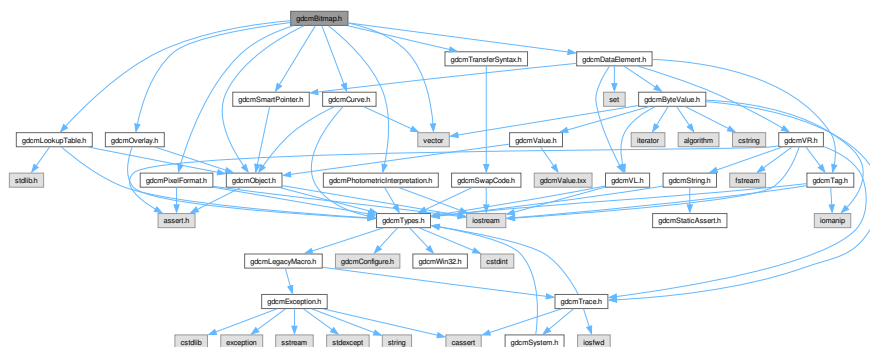
13.253 gdcmBitmap.h File Reference

```

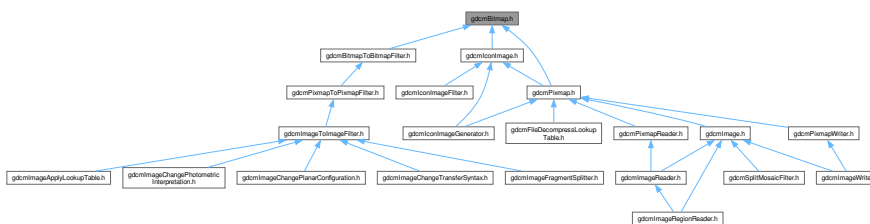
#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"

```

Include dependency graph for gdcMBitmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Bitmap`
Bitmap class.

Namespaces

- namespace `gdcm`

13.254 gdcmBitmap.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00006  00007
00008
```

```

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014 #ifndef GDCMBITMAP_H
00015 #define GDCMBITMAP_H
00016
00017 #include "gdcmObject.h"
00018 #include "gdcmCurve.h"
00019 #include "gdcmDataElement.h"
00020 // #include "gdcmIconImage.h"
00021 #include "gdcmLookupTable.h"
00022 #include "gdcmOverlay.h"
00023 #include "gdcmPhotometricInterpretation.h"
00024 #include "gdcmPixelFormat.h"
00025 #include "gdcmSmartPointer.h"
00026 #include "gdcmTransferSyntax.h"
00027
00028 #include <vector>
00029
00030 namespace gdcm
00031 {
00032
00033 class GDCM_EXPORT Bitmap : public Object
00034 {
00035 public:
00036     Bitmap();
00037     ~Bitmap() override;
00038     void Print(std::ostream &) const override;
00039
00040     virtual bool AreOverlaysInPixelData() const { return false; }
00041     virtual bool UnusedBitsPresentInPixelData() const { return false; }
00042
00043     unsigned int GetNumberOfDimensions() const;
00044     void SetNumberOfDimensions(unsigned int dim);
00045
00046     unsigned int GetPlanarConfiguration() const;
00047     void SetPlanarConfiguration(unsigned int pc);
00048
00049     bool GetNeedByteSwap() const
00050     {
00051         return NeedByteSwap;
00052     }
00053     void SetNeedByteSwap(bool b)
00054     {
00055         NeedByteSwap = b;
00056     }
00057
00058     void SetTransferSyntax(TransferSyntax const &ts) {
00059         TS = ts;
00060     }
00061     const TransferSyntax &GetTransferSyntax() const {
00062         return TS;
00063     }
00064     bool IsTransferSyntaxCompatible( TransferSyntax const & ts ) const;
00065     void SetDataElement(DataElement const &de) {
00066         PixelData = de;
00067     }
00068     const DataElement& GetDataElement() const { return PixelData; }
00069     DataElement& GetDataElement() { return PixelData; }
00070
00071     void SetLUT(LookupTable const &lut)
00072     {
00073         LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
00074     }
00075     const LookupTable &GetLUT() const
00076     {
00077         return *LUT;
00078     }
00079     LookupTable &GetLUT()
00080     {
00081         return *LUT;
00082     }
00083
00084     const unsigned int *GetDimensions() const;
00085     unsigned int GetDimension(unsigned int idx) const;
00086
00087     void SetColumns(unsigned int col) { SetDimension(0,col); }

```



```

00101 unsigned int GetColumns() const { return GetDimension(0); }
00102 void SetRows(unsigned int rows) { SetDimension(1,rows); }
00103 unsigned int GetRows() const { return GetDimension(1); }
00104 void SetDimensions(const unsigned int dims[3]);
00105 void SetDimension(unsigned int idx, unsigned int dim);
00107 const PixelFormat &GetPixelFormat() const
00108 {
00109     return PF;
00110 }
00111 PixelFormat &GetPixelFormat()
00112 {
00113     return PF;
00114 }
00115 void SetPixelFormat(PixelFormat const &pf)
00116 {
00117     PF = pf;
00118     PF.Validate();
00119 }
00120
00122 const PhotometricInterpretation &GetPhotometricInterpretation() const;
00123 void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00124
00125 bool IsEmpty() const { return Dimensions.empty(); }
00126 void Clear();
00127
00131 unsigned long GetBufferLength() const;
00132
00134 bool GetBuffer(char *buffer) const;
00135
00137 bool IsLossy() const;
00138
00140 void SetLossyFlag(bool f) { LossyFlag = f; }
00141
00142 protected:
00143 bool TryRAWCodec(char *buffer, bool &lossyflag) const;
00144 bool TryJPEGCodec(char *buffer, bool &lossyflag) const;
00145 bool TryPVRGCodec(char *buffer, bool &lossyflag) const;
00146 bool TryKAKADUCodec(char *buffer, bool &lossyflag) const;
00147 bool TryJPEGLSCodec(char *buffer, bool &lossyflag) const;
00148 bool TryJPEG2000Codec(char *buffer, bool &lossyflag) const;
00149 bool TryRLECodec(char *buffer, bool &lossyflag) const;
00150
00151 bool TryJPEGCodec2(std::ostream &os) const;
00152 bool TryJPEG2000Codec2(std::ostream &os) const;
00153
00154 bool GetBuffer2(std::ostream &os) const;
00155
00156 friend class PixmapReader;
00157 friend class ImageChangeTransferSyntax;
00158 // Function to compute the lossy flag based only on the image buffer.
00159 // Watch out that image can be lossy but in implicit little endian format...
00160 bool ComputeLossyFlag();
00161
00162 //private:
00163 protected:
00164 unsigned int PlanarConfiguration;
00165 unsigned int NumberOfDimensions;
00166 TransferSyntax TS;
00167 PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
00168 PhotometricInterpretation PI;
00169 // Mind dump: unsigned int is required here, since we are reading (0028,0008) Number Of Frames
00170 // which is VR::IS, so I cannot simply assumed that unsigned short is enough... :(
00171 std::vector<unsigned int> Dimensions; // Col/Row
00172 DataElement PixelData; // copied from 7fe0,0010
00173
00174 typedef SmartPointer<LookupTable> LUTPtr;
00175 LUTPtr LUT;
00176 // I believe the following 3 ivars can be derived from TS ...
00177 bool NeedByteSwap; // FIXME: remove me
00178 bool LossyFlag;
00179
00180 private:
00181 bool GetBufferInternal(char *buffer, bool &lossyflag) const;
00182 };
00183
00184 } // end namespace gdcm
00185
00186 #endif //GDCMBITMAP_H

```


13.257 gdcmCleaner.h File Reference

[illegible]

Classes

- class [gdcm::Cleaner](#)
[Cleaner](#).

Namespaces

- namespace [gdcm](#)

13.258 gdcmCleaner.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013
00014 #ifndef GDCMCLEANER_H
00015 #define GDCMCLEANER_H
00016
00017 #include "gdcmDPath.h"
00018 #include "gdcmFile.h"
00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmSubject.h"
00021
00022 namespace gdcm {
00030 class GDCM_EXPORT Cleaner : public Subject {
00031 public:
00032     Cleaner();
00033     ~Cleaner() override;
00034
00036     bool Empty(Tag const &t);
00037     bool Empty(PrivateTag const &pt);
00038     bool Empty(DPath const &dpath);
00039     bool Empty(VR const &vr);
00040
00041     bool Remove(Tag const &t);
00042     bool Remove(PrivateTag const &pt);
00043     bool Remove(DPath const &dpath);
00044     bool Remove(VR const &vr);
00045
00047     bool Scrub(Tag const &t);
00048     bool Scrub(PrivateTag const &pt);
00049     bool Scrub(DPath const &dpath);
00050     bool Scrub(VR const &vr);
00051
00052     // 8 Encoding of Coded Entry Data
00053     // https://dicom.nema.org/medical/dicom/current/output/chtml/part03/chapter_8.html
00054     typedef std::tuple<std::string, std::string, std::string> CodedEntryData;
00055
00057     bool ReplaceCodeMeaning(CodedEntryData const &ced);
00058
00060     bool Preserve(DPath const &dpath);
00061
00064     void RemoveAllMissingPrivateCreator(bool remove);
00065
00068     bool RemoveMissingPrivateCreator(Tag const &t);
00069
00071     void RemoveAllGroupLength(bool remove);

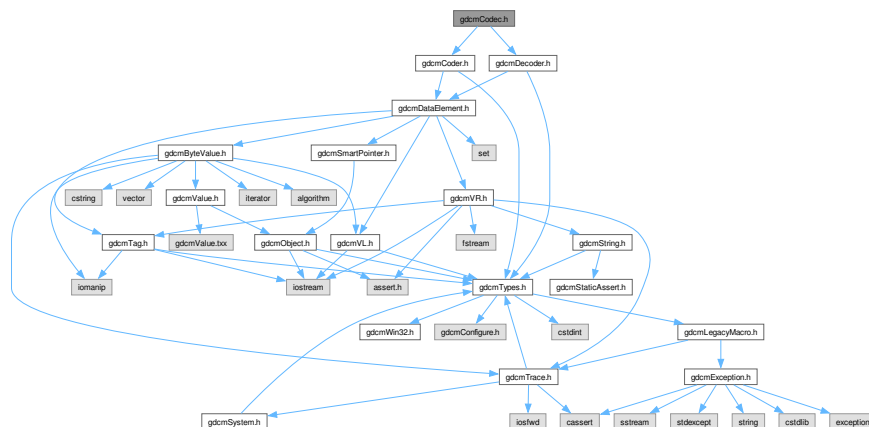
```

```

00072
00074 void RemoveAllIllegal(bool remove);
00075
00077 void EmptyWhenScrubFails(bool empty);
00078
00080 bool Clean();
00081
00083 void SetFile(const File &f) { F = f; }
00084 // const File &GetFile() const { return *F; }
00085 File &GetFile() { return *F; }
00086
00088 static SmartPointer<Cleaner> New() { return new Cleaner; }
00089
00090 private:
00091 // I would prefer to have a smart pointer to DataSet but DataSet does not
00092 // derive from Object...
00093 SmartPointer<File> F;
00094 struct impl;
00095 // PIMPL idiom
00096 impl *pimpl;
00097 };
00098
00099 } // end namespace gdcm
00100
00101 #endif // GDCMCLEANER_H

```

```
#include "gdcmlCoder.h"
#include "gdcmlDecoder.h"
Include dependency graph for gdcmlCodec.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Codec](#)
 [Codec](#) class.

Namespaces

- namespace [gdcm](#)

13.260 gdcmCodec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCODEC_H
00015  #define GDCMCODEC_H
00016
00017  #include "gdcmCoder.h"
00018  #include "gdcmDecoder.h"
00019
00020  namespace gdcm
00021  {
00022
00026  class GDCM_EXPORT Codec : public Coder, public Decoder
00027  {
00028  };
00029
00030  } // end namespace gdcm
00031
00032  #endif //GDCMCODEC_H

```

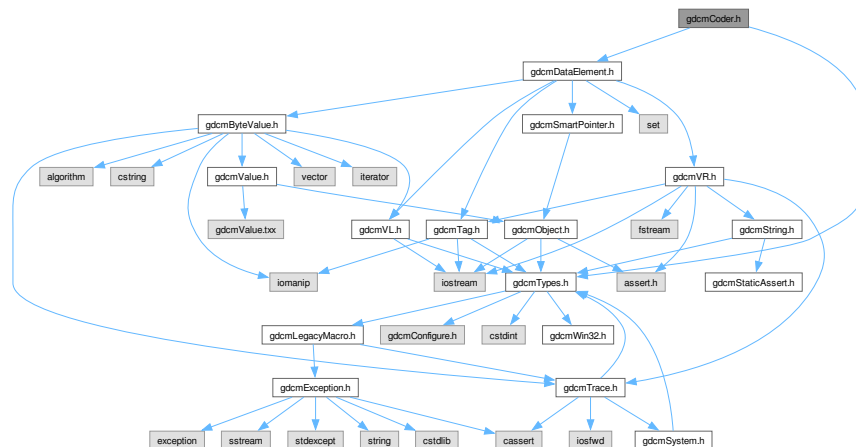
13.261 gdcmCoder.h File Reference

```

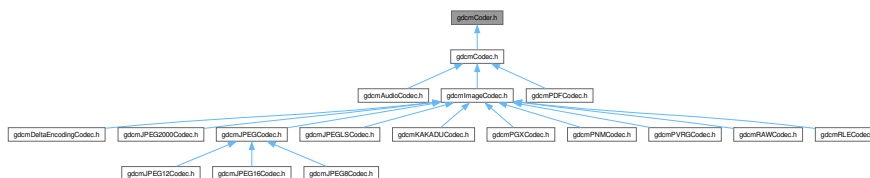
#include "gdcmTypes.h"
#include "gdcmDataElement.h"

```

Include dependency graph for gdcmlCoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcml::Coder`
`Coder`.

Namespaces

- namespace `gdcml`

13.262 gdcmlCoder.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
```

```

00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMCODER_H
00015 #define GDCMCODER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmDataElement.h" // FIXME
00019
00020 namespace gdcm
00021 {
00022
00023 class TransferSyntax;
00024 class DataElement;
00028 class GDCM_EXPORT Coder
00029 {
00030 public:
00031 virtual ~Coder() = default;
00032
00034 virtual bool CanCode(TransferSyntax const &) const = 0;
00035
00036 // Note: in / out are reserved keyword in C#. Change to in_ / out_
00037
00039 virtual bool Code(DataElement const &in_, DataElement &out_) { (void)in_; (void)out_; return false; }
00040 protected:
00041 virtual bool InternalCode(const char *bv, unsigned long len, std::ostream &os) { (void)bv;(void)os;(void)len;return false; }
00042 };
00043
00044 } // end namespace gdcm
00045
00046 #endif //GDCMCODER_H

```

13.263 gdcmConstCharWrapper.h File Reference

Classes

- class [gdcm::ConstCharWrapper](#)

Do not use me.

Namespaces

- namespace [gdcm](#)

13.264 gdcmConstCharWrapper.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012

```



```

00013
00014 =====*/
00014 #ifndef GDCMCONSTCHARWRAPPER_H
00015 #define GDCMCONSTCHARWRAPPER_H
00016
00017 namespace gdcm
00018 {
00019
00020 #error
00021
00022 /*
00023 * This class is a pure hack. Its only goal is to work around a bad bug in :
00024 * $ swig -version
00025 * SWIG Version 1.3.31
00026 *
00027 * See
00028 * - http://sourceforge.net/mailarchive/fo-
00029   rum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum_name=swig-user
00030 * As a side note there is also a problem with const reference to enum type:
00031 * - http://sourceforge.net/mailarchive/fo-
00032   rum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum_name=swig-user
00033 * And to keep track of an issue with swig here is the last one:
00034 *
00035 * - http://sourceforge.net/mailarchive/fo-
00036   rum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum_name=swig-user
00037 */
00038
00042 class ConstCharWrapper
00043 {
00044 public:
00045   ConstCharWrapper(const char *i=0):Internal(i) {}
00046   operator const char * () const { return Internal; }
00047 private:
00048   const char *Internal;
00049 };
00050
00051 } // end namespace gdcm
00052
00053 #endif //GDCMCONSTCHARWRAPPER_H

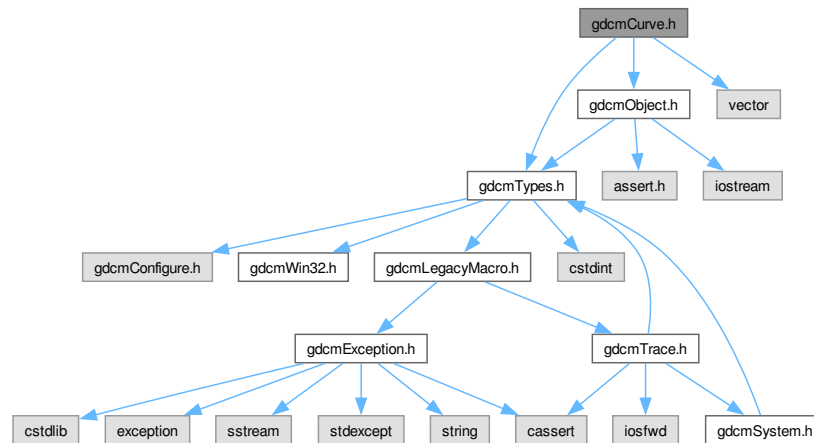
```

13.265 gdcmCurve.h File Reference

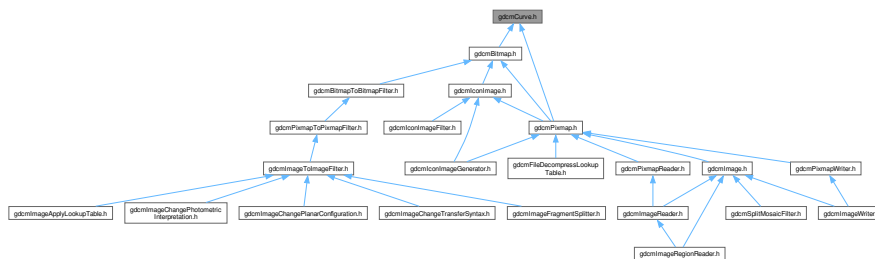
```

#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <vector>

```



gdmCurve.h



- class `gdc::Curve`
`Curve` class to handle element 50xx,3000 `Curve` Data.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```

00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMCURVE_H
00015 #define GDCMCURVE_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmObject.h"
00019
00020 #include <vector>
00021
00022 namespace gdcm
00023 {
00024
00025 class CurveInternal;
00026 class ByteValue;
00027 class DataSet;
00028 class DataElement;
00040 class GDCM_EXPORT Curve : public Object
00041 {
00042 public:
00043 Curve();
00044 ~Curve() override;
00045 void Print(std::ostream &) const override;
00046
00047 void GetAsPoints(float *array) const;
00048
00049 static unsigned int GetNumberOfCurves(DataSet const & ds);
00050
00051 // Update curve data from dataelement de:
00052 void Update(const DataElement & de);
00053
00054 void SetGroup(unsigned short group);
00055 unsigned short GetGroup() const;
00056 void SetDimensions(unsigned short dimensions);
00057 unsigned short GetDimensions() const;
00058 void SetNumberOfPoints(unsigned short numberofpoints);
00059 unsigned short GetNumberOfPoints() const;
00060 void SetTypeOfData(const char *typeofdata);
00061 const char *GetTypeOfData() const;
00062 // See PS 3.3 - 2004 - C.10.2.1.1 Type of data
00063 const char *GetTypeOfDataDescription() const;
00064 void SetCurveDescription(const char *curvedescription);
00065 void SetDataValueRepresentation(unsigned short datavaluerepresentation);
00066 unsigned short GetDataValueRepresentation() const;
00067 void SetCurveDataDescriptor(const uint16_t * values, size_t num);
00068 std::vector<unsigned short> const &GetCurveDataDescriptor() const;
00069 void SetCoordinateStartValue( unsigned short v );
00070 void SetCoordinateStepValue( unsigned short v );
00071
00072 void SetCurve(const char *array, unsigned int length);
00073
00074 bool IsEmpty() const;
00075
00076 void Decode(std::istream &is, std::ostream &os);
00077
00078 Curve(Curve const &ov);
00079 private:
00080 double ComputeValueFromStartAndStep(unsigned int idx) const;
00081 CurveInternal *Internal;
00082 };
00083
00084 } // end namespace gdcm
00085
00086 #endif //GDCMCURVE_H

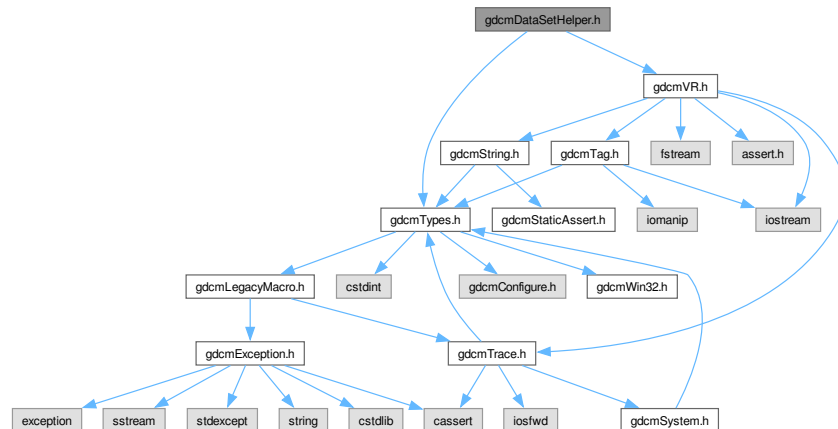
```

13.267 gdcmDataSetHelper.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmDataSetHelper.h:



Classes

- class [gdcm::DataSetHelper](#)
[DataSetHelper](#) (internal class, not intended for user level).

Namespaces

- namespace [gdcm](#)

13.268 gdcmDataSetHelper.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDATASETHelper_H
00015 #define GDCMDATASETHelper_H
00016

```

```

00017 #include "gdcmTypes.h"
00018 #include "gdcmVR.h"
00019
00020 namespace gdcm
00021 {
00022   class DataSet;
00023   class File;
00024   class Tag;
00025   class SequenceOfItems;
00026
00030   class GDCM_EXPORT DataSetHelper
00031   {
00032   public:
00035     static VR ComputeVR(File const & file, DataSet const &ds, const Tag& tag);
00036
00037     //static SequenceOfItems* ComputeSQFromByteValue(File const & file, DataSet const &ds, const Tag &tag);
00038
00039   protected:
00040   };
00041
00042 } // end namespace gdcm
00043
00044 #endif // GDCMDATASETHELPER_H

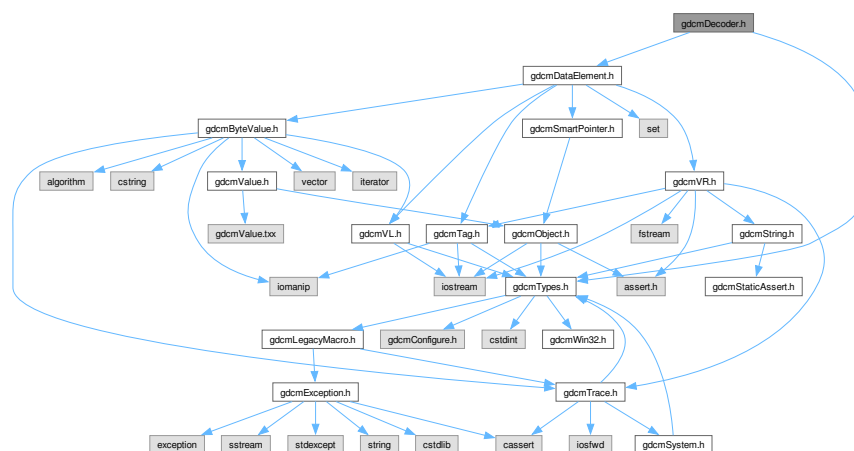
```

13.269 gdcmDecoder.h File Reference

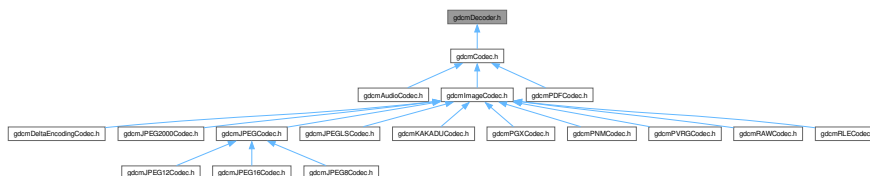
```
#include "gdcmTypes.h"
```

```
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmDecoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Decoder`
`Decoder`.

Namespaces

- namespace `gdcm`

13.270 `gdcmDecoder.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013
00014
00015 #ifndef GDCMDECODER_H
00016 #define GDCMDECODER_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmDataElement.h" // FIXME
00020
00021 namespace gdcm
00022 {
00023
00024 class TransferSyntax;
00025 class DataElement;
00026 class GDCM_EXPORT Decoder
00027 {
00028 public:
00029     virtual ~Decoder() = default;
00030
00031     virtual bool CanDecode(TransferSyntax const &) const = 0;
00032
00033     virtual bool Decode(DataElement const &, DataElement &) { return false; }
00034 protected:
00035     virtual bool DecodeByStreams(std::istream &, std::ostream &) { return false; }
00036 };
00037
00038 } // end namespace gdcm
00039
00040 #endif //GDCMDECODER_H

```


13.274 gdcmDICOMDIR.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDICOMDIR_H
00015 #define GDCMDICOMDIR_H
00016
00017 #include <utility>
00018 #include "gdcmFileSet.h"
00019
00020 namespace gdcm
00021 {
00022     class GDCM_EXPORT DICOMDIR
00023     {
00024     public:
00025         DICOMDIR() = default;
00026         DICOMDIR(FileSet fs):_FS(std::move(std::move(fs))) {}
00027     private:
00028         FileSet _FS;
00029         //13 sept 2010 mmr-- added the underscore to FS to compile under Sunos gcc
00030     };
00031 } // end namespace gdcm
00032
00033 #endif //GDCMDICOMDIR_H

```

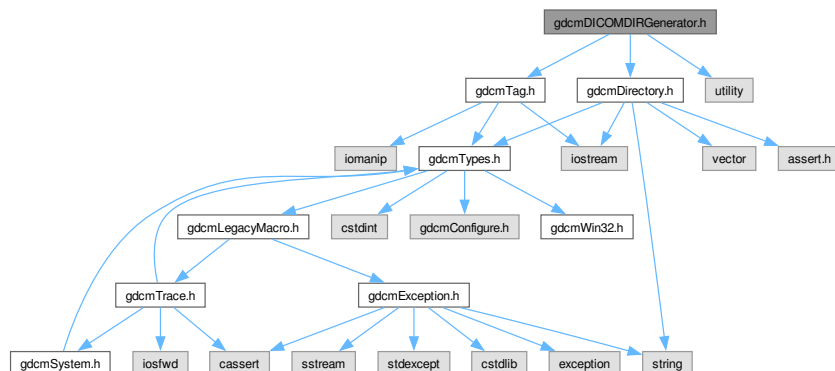
13.275 gdcmDICOMDIRGenerator.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>

```

Include dependency graph for gdcmDICOMDIRGenerator.h:



Classes

- class [gdcm::DICOMDIRGenerator](#)
[DICOMDIRGenerator](#) class.

Namespaces

- namespace [gdcm](#)

13.276 gdcmDICOMDIRGenerator.h

[Go to the documentation of this file.](#)

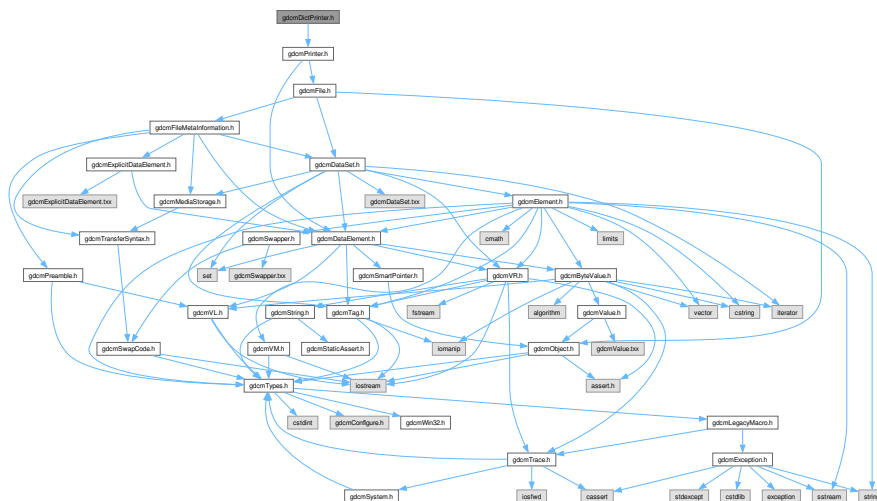
```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013  #ifndef GDCMDICOMDIRGENERATOR_H
00014  #define GDCMDICOMDIRGENERATOR_H
00015
00016  #include "gdcmDirectory.h"
00017  #include "gdcmTag.h"
00018  #include <utility> // std::pair
00019
00020  namespace gdcm
00021  {
00022  {
00023  class File;
00024  class Scanner;
00025  class SequenceOfItems;
00026  class VL;
00027  class DICOMDIRGeneratorInternal;
00028
00029  class GDCM_EXPORT DICOMDIRGenerator
00030  {
00031  public:
00032  typedef Directory::FilenameType FilenameType;
00033  typedef Directory::FilenameType FilenameType;
00034  DICOMDIRGenerator();
00035  ~DICOMDIRGenerator();
00036
00037  void SetFilenames( FilenameType const & fns );
00038
00039  void SetRootDirectory( FilenameType const & root );
00040
00041  void SetDescriptor( const char *d );
00042
00043  bool Generate();
00044
00045  void SetFile(const File& f);
00046  File &GetFile();
00047
00048  protected:
00049  Scanner &GetScanner();
00050  bool AddPatientDirectoryRecord();
00051  bool AddStudyDirectoryRecord();
00052  bool AddSeriesDirectoryRecord();
00053  bool AddImageDirectoryRecord();
00054
00055  private:

```

13.277 gdcDictPrinter.h File Reference

Include dependency graph for `gdcmDictPrinter.h`:



- class `gdcmm::DictPrinter`
`DictPrinter` class.

- namespace `gdcm`

13.278 gdcmDictPrinter.h

[Go to the documentation of this file.](#)

```

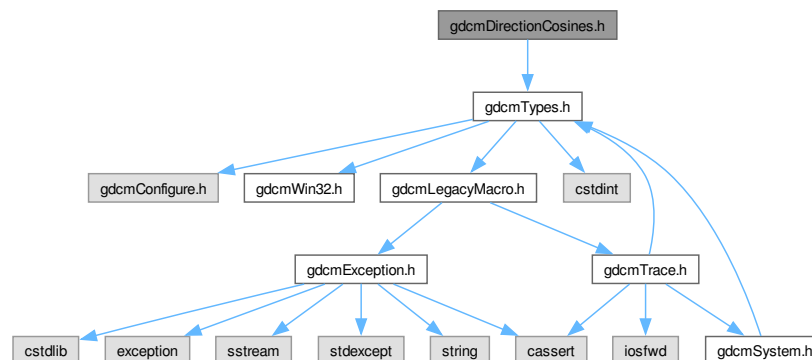
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDICTPRINTER_H
00015 #define GDCMDICTPRINTER_H
00016
00017 #include "gdcmPrinter.h"
00018
00019 namespace gdcm
00020 {
00021
00025 // It's a sink there is no output
00026 class GDCM_EXPORT DictPrinter : public Printer
00027 {
00028 public:
00029   DictPrinter();
00030   ~DictPrinter() = default;
00031
00032   void Print(std::ostream& os);
00033
00034 protected:
00035   void PrintDataElement2(std::ostream& os, const DataSet &ds, const DataElement &ide);
00036   void PrintDataSet2(std::ostream& os, const DataSet &ds);
00037 };
00038
00039 } // end namespace gdcm
00040
00041 #endif //GDCMDICTPRINTER_H

```

13.279 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDirectionCosines.h:



Classes

- class `gdcm::DirectionCosines`
class to handle `DirectionCosines`

Namespaces

- namespace `gdcm`

13.280 gdcmDirectionCosines.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMDIRECTIONCOSINES_H
00015  #define GDCMDIRECTIONCOSINES_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021
00025  class GDCM_EXPORT DirectionCosines
00026  {
00027  public:
00028  DirectionCosines();
00029  DirectionCosines(const double dircos[6]);
00030  // Cannot get the following signature to be wrapped with swig...
00031  //DirectionCosines(const double *dircos = 0 );
00032  ~DirectionCosines() = default;
00033
00035  void Print(std::ostream &) const;
00036
00038  void Cross(double z[3]) const;
00039
00041  double Dot() const;
00042
00044  static double Dot(const double x[3], const double y[3]);
00045
00047  void Normalize();
00048
00050  static void Normalize(double v[3]);
00051
00053  static double Norm(const double v[3]);
00054
00056  operator const double* () const { return Values; }
00057
00059  bool IsValid() const;
00060
00063  bool SetFromString(const char *str);
00064
00066  double CrossDot(DirectionCosines const &dc) const;
00067
00069  double ComputeDistAlongNormal(const double ipp[3]) const;
00070
00071  private:

```

```

00072  double Values[6];
00073  };
00074
00075 } // end namespace gdcmm
00076
00077 #endif //GDCMDIRECTIONCOSINES_H

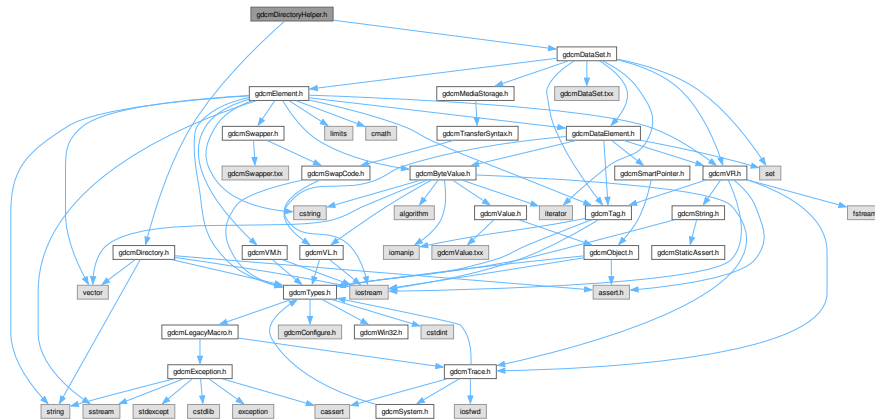
```

13.281 gdcmmDirectoryHelper.h File Reference

```
#include "gdcmmDirectory.h"
```

```
#include "gdcmmDataSet.h"
```

Include dependency graph for gdcmmDirectoryHelper.h:



Classes

- class [gdcmm::DirectoryHelper](#)
[DirectoryHelper](#).

Namespaces

- namespace [gdcmm](#)

13.282 gdcmmDirectoryHelper.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even

```

```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00015 #include "gdcmDirectory.h"
00016 #include "gdcmDataSet.h"
00017
00018 namespace gdcm
00019 {
00020
00035 class GDCM_EXPORT DirectoryHelper
00036 {
00037 public:
00038     //returns all series UUIDs in a given directory that match a particular SOP Instance UID
00039     static Directory::FileNamesType GetSeriesUUIDsBySOPClassUID(const std::string& inDirectory,
00040         const std::string& inSOPClassUID);
00041
00042     //specific implementations of the SOPClassUID grabber, so you don't have to
00043     //remember the SOP Class UUIDs of CT or MR images.
00044     static Directory::FileNamesType GetCTImageSeriesUUIDs(const std::string& inDirectory);
00045     static Directory::FileNamesType GetMRImageSeriesUUIDs(const std::string& inDirectory);
00046     static Directory::FileNamesType GetRTStructSeriesUUIDs(const std::string& inDirectory);
00047
00048     //given a directory and a series UID, provide all filenames with that series UID.
00049     static Directory::FileNamesType GetFilenamesFromSeriesUUIDs(const std::string& inDirectory,
00050         const std::string& inSeriesUID);
00051
00052     //given a series UID, load all the images associated with that series UID
00053     //these images will be IPP sorted, so that they can be used for gathering all
00054     //the necessary information for generating an RTStruct
00055     //this function should be called by the writer once, if the writer's dataset
00056     //vector is empty. Make sure to have a new writer for new rtstructs.
00057     static std::vector<DataSet> LoadImageFromFiles(const std::string& inDirectory,
00058         const std::string& inSeriesUID);
00059
00060     //When writing RTStructs, each contour will have z position defined.
00061     //use that z position to determine the SOPInstanceUID for that plane.
00062     static std::string RetrieveSOPInstanceUIDFromZPosition(double inZPos,
00063         const std::vector<DataSet>& inDS);
00064
00065     //When writing RTStructs, the frame of reference is done by planes to start with
00066     static std::string RetrieveSOPInstanceUIDFromIndex(int inIndex,
00067         const std::vector<DataSet>& inDS);
00068
00069     //each plane needs to know the SOPClassUID, and that won't change from image to image
00070     //so, retrieve this once at the start of writing.
00071     static std::string GetSOPClassUID(const std::vector<DataSet>& inDS);
00072
00073     //retrieve the frame of reference from the set of datasets
00074     static std::string GetFrameOfReference(const std::vector<DataSet>& inDS);
00075
00076     //both the image and polydata readers use these functions to get std::strings
00077     static std::string GetStringValueFromTag(const Tag& t, const DataSet& ds);
00078 };
00079
00080 }

```

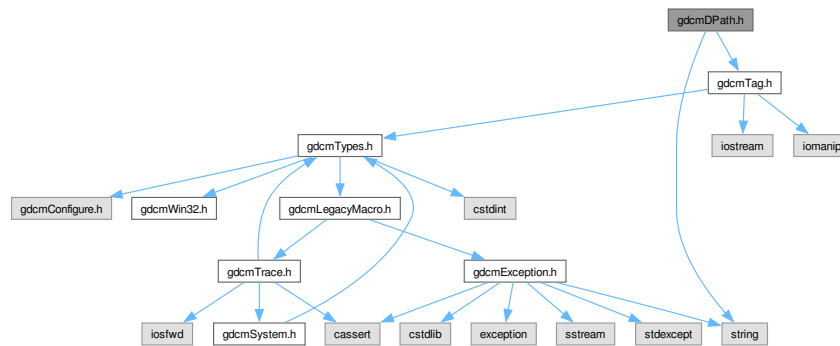
13.283 gdcmDPath.h File Reference

```

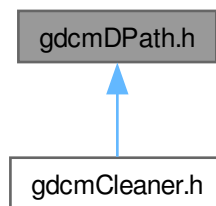
#include "gdcmTag.h"
#include <string>

```

Include dependency graph for `gdcmDPath.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::DPath`
class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation <https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA>

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DPath &val)`

13.284 gdcmDPath.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDPATH_H
00015 #define GDCMDPATH_H
00016
00017 #include "gdcmTag.h"
00018 #include <string>
00019
00020 namespace gdcm {
00021
00022 class GDCM_EXPORT DPath {
00023 friend std::ostream &operator«(std::ostream &_os, const DPath &_val);
00024
00025 public:
00026 DPath();
00027 ~DPath();
00028 void Print(std::ostream &) const;
00029 bool operator<(const DPath &rhs) const;
00030
00031 bool ConstructFromString(const char *path);
00032
00033 bool Match(DPath const &other) const;
00034
00035 static bool IsValid(const char *path);
00036
00037 private:
00038 std::string Path;
00039 };
00040
00041 inline std::ostream &operator«(std::ostream &os, const DPath &val) {
00042 os « val.Path;
00043 return os;
00044 }
00045
00046 } // end namespace gdcm
00047
00048 #endif // GDCMDPATH_H

```



```

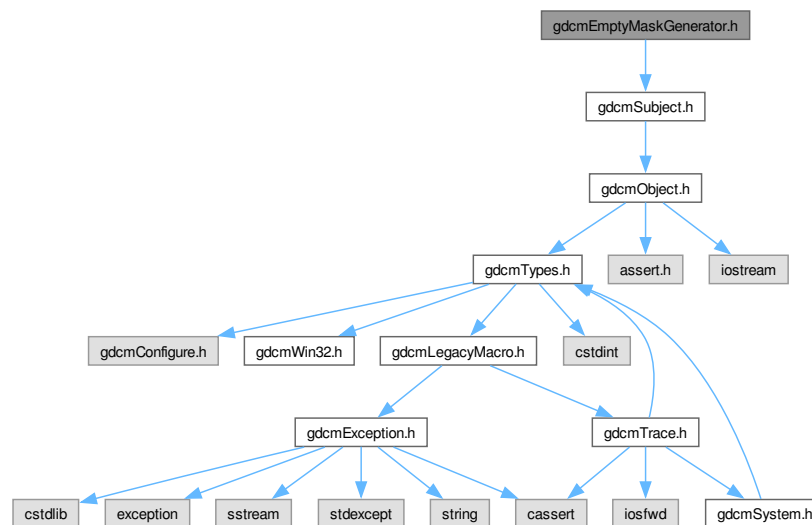
00017 #include "gdcmPrinter.h"
00018
00019 namespace gdcm
00020 {
00021
00022 // It's a sink there is no output
00029 class GDCM_EXPORT Dumper : public Printer
00030 {
00031 public:
00032     Dumper() { PrintStyle = CONDENSED_STYLE; }
00033     ~Dumper() = default;
00034 };
00035
00036 } // end namespace gdcm
00037
00038 #endif //GDCMDUMPER_H

```

13.287 gdcmEmptyMaskGenerator.h File Reference

```
#include "gdcmSubject.h"
```

Include dependency graph for gdcmEmptyMaskGenerator.h:



Classes

- class [gdcm::EmptyMaskGenerator](#)

EmptyMaskGenerator Main class to generate a Empty Mask [Series](#) from an input [Series](#). This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM [Series](#) within the same input directory.

Namespaces

- namespace [gdcm](#)

13.288 gdcmEmptyMaskGenerator.h

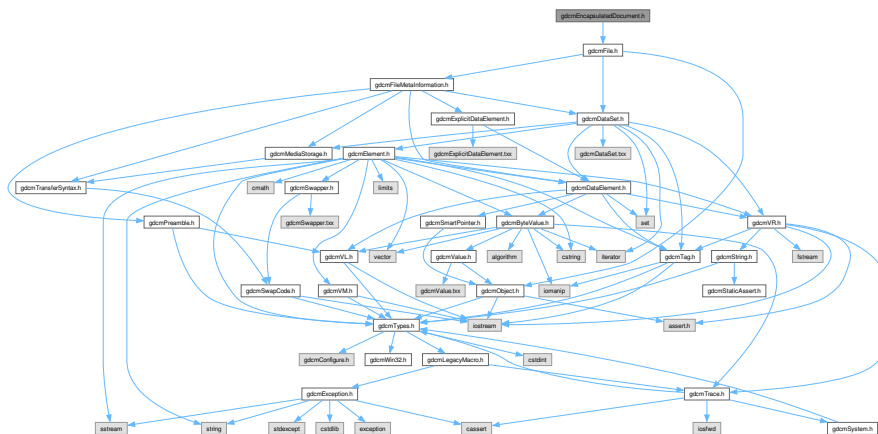
[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMEMPTYMASKGENERATOR_H
00015 #define GDCMEMPTYMASKGENERATOR_H
00016
00017 #include "gdcmSubject.h"
00018
00019 namespace gdcm {
00020 class GDCM_EXPORT EmptyMaskGenerator
00021 {
00022 public:
00023     EmptyMaskGenerator();
00024     ~EmptyMaskGenerator();
00025
00026     enum SOPClassUIDMode {
00027         UseOriginalSOPClassUID = 0, // default
00028         UseGrayscaleSecondaryImageStorage
00029     };
00030
00031     void SetSOPClassUIDMode( SOPClassUIDMode mode );
00032
00033     void SetInputDirectory( const char * dirname );
00034
00035     void SetOutputDirectory( const char * dirname );
00036
00037     bool Execute();
00038
00039 private:
00040     struct impl;
00041     // PIMPL idiom
00042     impl* pimpl;
00043 };
00044 } // end namespace gdcm
00045 #endif //GDCMEMPTYMASKGENERATOR_H

```

```
#include "gdcmFile.h"
Include dependency graph for gdcmEncapsulatedDocument.h:
```



- class `gdcm::EncapsulatedDocument`
`EncapsulatedDocument`.

- namespace `gdcm`

Go to the documentation of this file.

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013 #ifndef GDCMENCAPSULATEDDOCUMENT_H
00014 #define GDCMENCAPSULATEDDOCUMENT_H
00015
00016 #include "gdcmFile.h"
00017
00018 namespace gdcm
00019

```

```

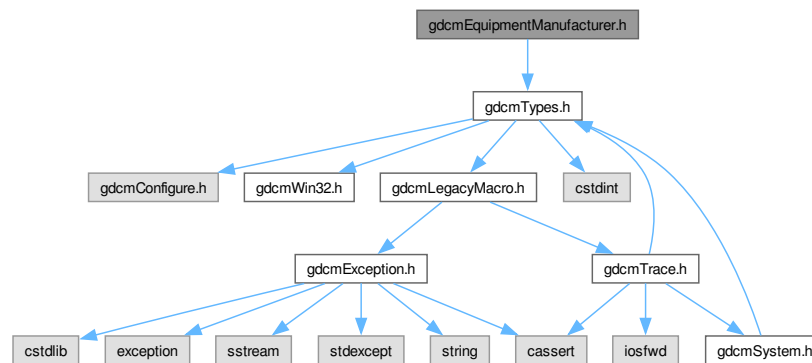
00020 {
00024 class GDCM_EXPORT EncapsulatedDocument
00025 {
00026 public:
00027     EncapsulatedDocument() = default;
00028
00029 private:
00030 };
00031
00032 } // end namespace gdc
00033
00034 #endif //GDCMENCAPSULATEDDOCUMENT_H

```

13.291 gdcEquipmentManufacturer.h File Reference

#include "gdcTypes.h"

Include dependency graph for gdcEquipmentManufacturer.h:



Classes

- class [gdc::EquipmentManufacturer](#)

Namespaces

- namespace [gdc](#)

13.292 gdcEquipmentManufacturer.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004 Copyright (c) 2006-2011 Mathieu Malaterre
00005 All rights reserved.

```

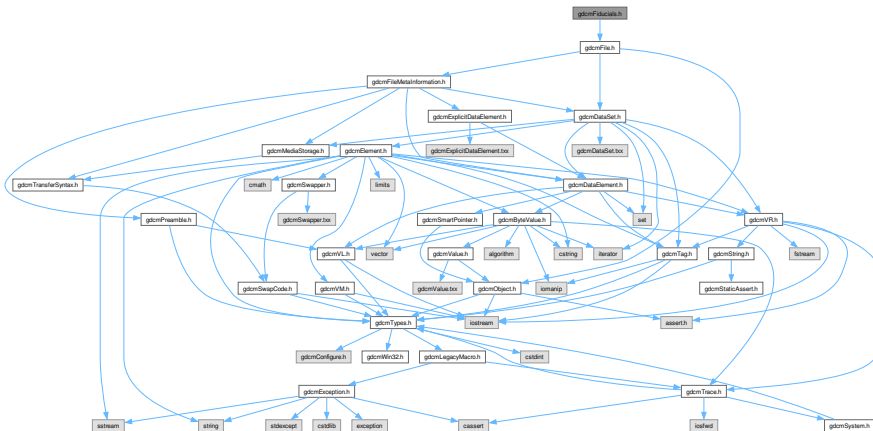
```
00007 See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
```

```
00014 ===== */
00014 #ifndef GDCMEQUIPMENTMANUFACTURER_H
00015 #define GDCMEQUIPMENTMANUFACTURER_H
00016
00017 #include "gdcmtypes.h"
00018
00019 namespace gdcmt {
00020
00021 class DataSet;
00022 class GDCM_EXPORT EquipmentManufacturer {
00023 public:
00024     typedef enum {
00025         UNKNOWN = 0,
00026         AGFA,
00027         FUJI,
00028         GEMS,
00029         HITACHI,
00030         KODAK,
00031         MARCONI,
00032         PMS,
00033         SAMSUNG,
00034         SIEMENS,
00035         TOSHIBA,
00036         UIH
00037     } Type;
00038
00039     static Type Compute(DataSet const &ds);
00040
00041     static const char *TypeToString(Type type);
00042
00043 private:
00044     static EquipmentManufacturer::Type GuessFromPrivateAttributes(
00045         DataSet const &ds);
00046 };
00047
00048 } // end namespace gdcmt
00049
00050 #endif // GDCMEQUIPMENTMANUFACTURER_H
```

13.293 gdcmFiducials.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcMfiducials.h:



Classes

- class [gdcm::Fiducials](#)
[Fiducials](#).

Namespaces

- namespace [gdcm](#)

13.294 gdcmFiducials.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMFIDUCIALS_H
00015  #define GDCMFIDUCIALS_H
00016
00017  #include "gdcmFile.h"
00018
00019  namespace gdcm
00020  {
00021
00024  class GDCM_EXPORT Fiducials
00025  {
00026  public:
00027      Fiducials() = default;
00028
00029  private:
00030  };
00031
00032  } // end namespace gdcm
00033
00034  #endif //GDCMFIDUCIALS_H

```

13.295 gdcmFileAnonymizer.h File Reference

```

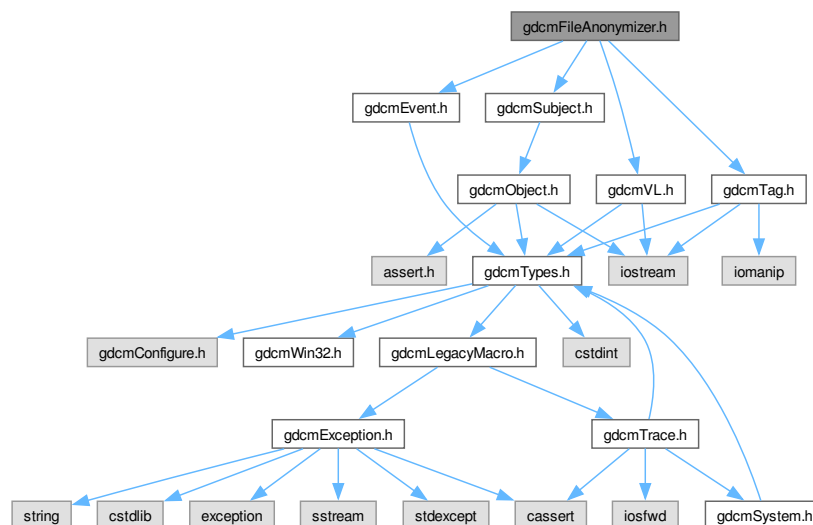
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"

```



```
#include "gdcmVL.h"
```

Include dependency graph for gdcmFileAnonymizer.h:



Classes

- class [gdcm::FileAnonymizer](#)
[FileAnonymizer](#).

Namespaces

- namespace [gdcm](#)

13.296 gdcmFileAnonymizer.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMFILEANONYMIZER_H
00015 #define GDCMFILEANONYMIZER_H
00016

```

```

00017 #include "gdcmSubject.h"
00018 #include "gdcmEvent.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmVL.h"
00021
00022 namespace gdcm
00023 {
00024 class FileAnonymizerInternals;
00025
00047 class GDCM_EXPORT FileAnonymizer : public Subject
00048 {
00049 public:
00050   FileAnonymizer();
00051   ~FileAnonymizer() override;
00052
00055   void Empty( Tag const &t );
00056
00058   void Remove( Tag const &t );
00059
00063   void Replace( Tag const &t, const char *value_str );
00064
00067   void Replace( Tag const &t, const char *value_data, VL const &vl );
00068
00070   void SetInputFileName(const char *filename_native);
00071
00073   void SetOutputFileName(const char *filename_native);
00074
00076   bool Write();
00077
00078 private:
00079   bool ComputeEmptyTagPosition();
00080   bool ComputeRemoveTagPosition();
00081   bool ComputeReplaceTagPosition();
00082   FileAnonymizerInternals *Internals;
00083 };
00084
00085 } // end namespace gdcm
00086
00087 #endif //GDCMFILEANONYMIZER_H

```

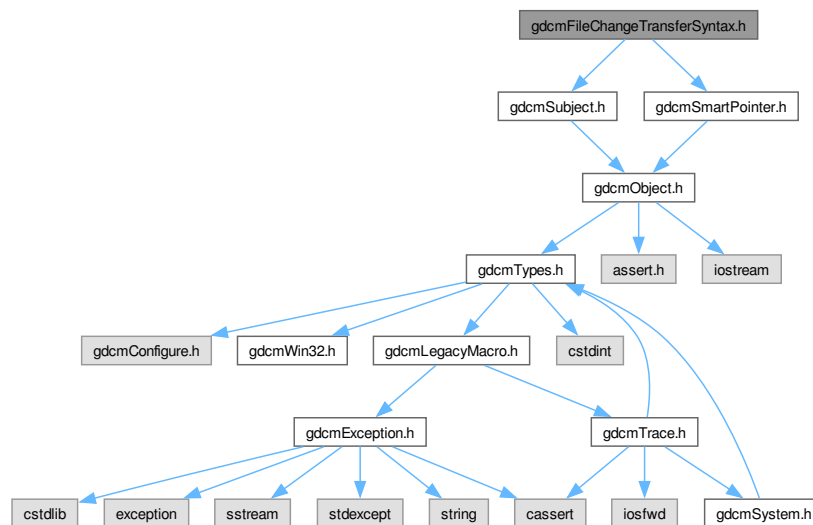
13.297 gdcmFileChangeTransferSyntax.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class `gdcm::FileChangeTransferSyntax`
`FileChangeTransferSyntax`.

Namespaces

- namespace `gdcm`

13.298 gdcmFileChangeTransferSyntax.h

[Go to the documentation of this file.](#)

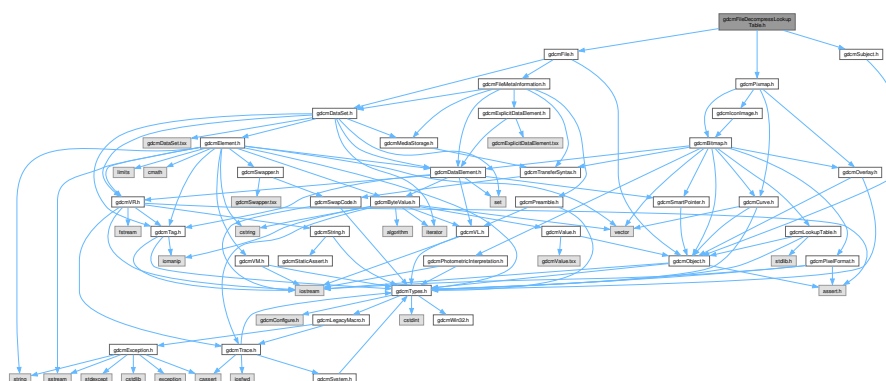
```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMFILECHANGETRANSFERSYNTAX_H
00015  #define GDCMFILECHANGETRANSFERSYNTAX_H
00016  00017  #include "gdcmSubject.h"
00018  #include "gdcmSmartPointer.h"

```

13.299 [gdcmFileDecompressLookupTable.h](#) File Reference

Include dependency graph for gdcMFileDecompressLookupTable.h:



- class `gdcm::FileDecompressLookupTable`
`FileDecompressLookupTable` class.

Namespaces

- namespace [gdcm](#)

13.300 gdcmFileDecompressLookupTable.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMFILEDECOMPRESSLOOKUPTABLE_H
00015  #define GDCMFILEDECOMPRESSLOOKUPTABLE_H
00016  00017  #include "gdcmSubject.h"
00018  #include "gdcmFile.h"
00019  #include "gdcmPixmap.h"
00020  00021  namespace gdcm
00022  {
00023  00024  class DataElement;
00030  class GDCM_EXPORT FileDecompressLookupTable : public Subject
00031  {
00032  public:
00033  FileDecompressLookupTable() = default;
00034  ~FileDecompressLookupTable() override = default;
00035  00037  bool Change();
00038  00040  void SetFile(const File& f) { F = f; }
00041  File &GetFile() { return *F; }
00042  00043  const Pixmap& GetPixmap() const { return *PixelData; }
00044  Pixmap& GetPixmap() { return *PixelData; }
00045  void SetPixmap(Pixmap const &img) { PixelData = img; }
00046  00047  protected:
00048  00049  private:
00050  SmartPointer<File> F;
00051  SmartPointer<Pixmap> PixelData;
00052  };
00053  00054  } // end namespace gdcm
00055  00056  #endif //GDCMFILEDECOMPRESSLOOKUPTABLE_H

```



```
00020 {
00021
00022 class FileDerivationInternals;
00023 class DataSet;
00039 class GDCM_EXPORT FileDerivation
00040 {
00041 public:
00042 FileDerivation();
00043 ~FileDerivation();
00044
00049 bool AddReference(const char *referencedsopclassuid, const char *referencedsopinstanceuid);
00050
00051 // CID 7202 Source Image Purposes of Reference
00052 // { "DCM",121320,"Uncompressed predecessor" },
00053
00055 void SetPurposeOfReferenceCodeSequenceCodeValue(unsigned int codevalue);
00056
00057 // CID 7203 Image Derivation
00058 // { "DCM",113040,"Lossy Compression" },
00059
00061 void SetDerivationCodeSequenceCodeValue(unsigned int codevalue);
00062
00064 void SetDerivationDescription( const char *dd );
00065
00069 void SetAppendDerivationHistory(bool b);
00070
00072 bool Derive();
00073
00075 void SetFile(const File& f) { F = f; }
00076 File &GetFile() { return *F; }
00077 const File &GetFile() const { return *F; }
00078
00079 protected:
00080 bool AddDerivationDescription();
00081 bool AddSourceImageSequence();
00082 bool AddPurposeOfReferenceCodeSequence(DataSet &ds);
00083
00084 private:
00085 SmartPointer<File> F;
00086 FileDerivationInternals *Internals;
00087 };
00088
00094
00095
00096 } // end namespace gdcm
00097
00098 #endif //GDCMFILEDERIVATION_H
```



```

00020 {
00021   class Dicts;
00022
00038   class GDCM_EXPORT FileExplicitFilter
00039   {
00040   public:
00041     FileExplicitFilter():F(new
00042     File),ChangePrivateTags(false),UseVRUN(true),RecomputeItemLength(false),RecomputeSequenceLength(false) {}
00042     ~FileExplicitFilter() = default;
00043
00045     void SetChangePrivateTags(bool b) { ChangePrivateTags = b;}
00046
00048     void SetUseVRUN(bool b) { UseVRUN = b; }
00049
00051     void SetRecomputeItemLength(bool b);
00052     void SetRecomputeSequenceLength(bool b);
00053
00055
00057     bool Change();
00058
00060     void SetFile(const File& f) { F = f; }
00061     File &GetFile() { return *F; }
00062
00063   protected:
00064     bool ProcessDataSet(DataSet &ds, Dicts const &dicts);
00065     bool ChangeFMI();
00066
00067   private:
00068     SmartPointer<File> F;
00069     bool ChangePrivateTags;
00070     bool UseVRUN;
00071     bool RecomputeItemLength;
00072     bool RecomputeSequenceLength;
00073   };
00074
00075
00076 } // end namespace gdcm
00077
00078 #endif //GDCMFILEEXPLICITFILTER_H

```

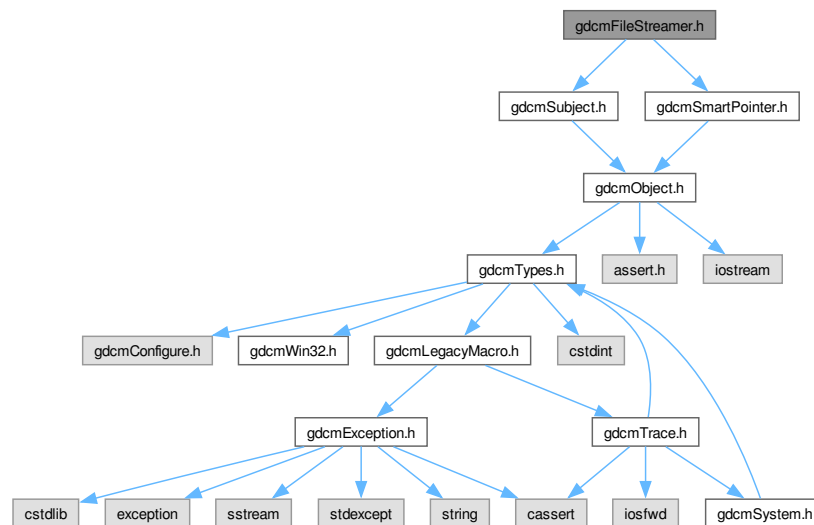
13.305 gdcmFileStreamer.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for `gdcmFileStreamer.h`:



Classes

- class `gdcm::FileStreamer`
`FileStreamer`.

Namespaces

- namespace `gdcm`

13.306 `gdcmFileStreamer.h`

[Go to the documentation of this file.](#)

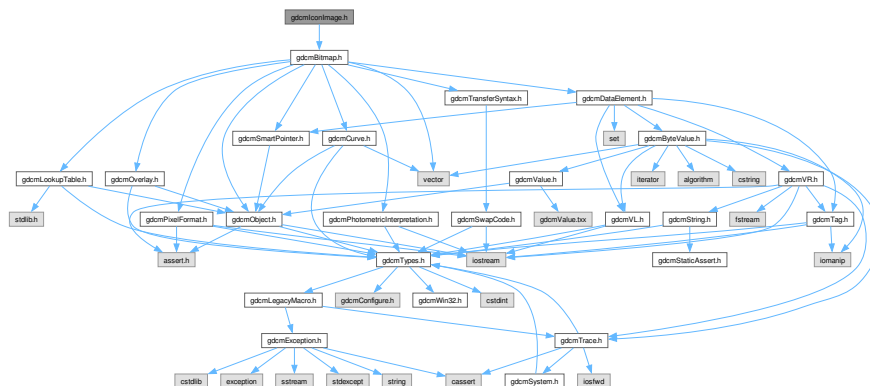
```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMFILESTREAMER_H
00015  #define GDCMFILESTREAMER_H
00016  00017  #include "gdcmSubject.h"
00018  #include "gdcmSmartPointer.h"

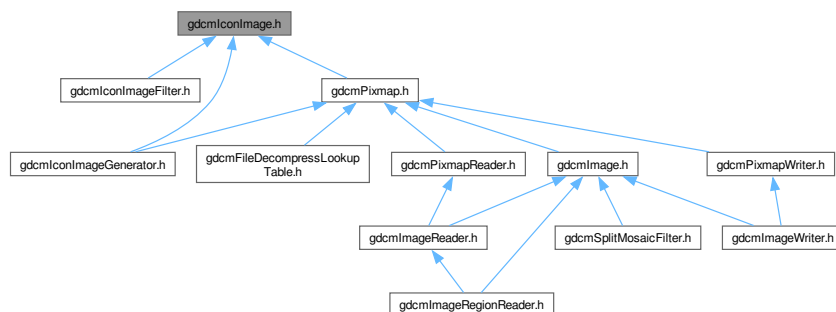
```

13.307 gdcmIconImage.h File Reference

Include dependency graph for `gdcmlconImage.h`:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace [gdcm](#)

Typedefs

- typedef [Bitmap](#) [gdcm::IconImage](#)

13.308 gdcmIconImage.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMICONIMAGE_H
00015 #define GDCMICONIMAGE_H
00016
00017 #if 0
00018 #include "gdcmObject.h"
00019 #include "gdcmDataElement.h"
00020 #include "gdcmPhotometricInterpretation.h"
00021 #include "gdcmPixelFormat.h"
00022 #include "gdcmTransferSyntax.h"
00023
00024 #include <vector>
00025
00026 namespace gdcm
00027 {
00028
00029 class GDCM_EXPORT IconImage : public Object
00030 {
00031

```

```

00034 public:
00035   IconImage();
00036   ~IconImage();
00037   void Print(std::ostream &c) const {}
00038
00040   void SetTransferSyntax(TransferSyntax const &ts) {
00041     TS = ts;
00042   }
00043   const TransferSyntax &GetTransferSyntax() const {
00044     return TS;
00045   }
00046   void SetDataElement(DataElement const &de) {
00047     PixelData = de;
00048   }
00049   const DataElement& GetDataElement() const { return PixelData; }
00050
00051   void SetColumns(unsigned int col) { SetDimension(0,col); }
00052   void SetRows(unsigned int rows) { SetDimension(1,rows); }
00053   void SetDimension(unsigned int idx, unsigned int dim);
00054   int GetColumns() const { return Dimensions[0]; }
00055   int GetRows() const { return Dimensions[1]; }
00056   // Get/Set PixelFormat
00057   const PixelFormat &GetPixelFormat() const
00058   {
00059     return PF;
00060   }
00061   void SetPixelFormat(PixelFormat const &pf)
00062   {
00063     PF = pf;
00064   }
00065
00066   const PhotometricInterpretation &GetPhotometricInterpretation() const;
00067   void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00068
00069   bool IsEmpty() const { return Dimensions.size() == 0; }
00070   void Clear();
00071
00072   bool GetBuffer(char *buffer) const;
00073
00074 private:
00075   TransferSyntax TS;
00076   PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
00077   PhotometricInterpretation PI;
00078   std::vector<unsigned int> Dimensions; // Col/Row
00079   std::vector<double> Spacing; // PixelAspectRatio ?
00080   DataElement PixelData; // copied from 7fe0,0010
00081   static const unsigned int NumberOfDimensions = 2;
00082 };
00083
00084 } // end namespace gdcm
00085 #endif
00086 #include "gdcmBitmap.h"
00087
00088 namespace gdcm
00089 {
00090   //class GDCM_EXPORT IconImage : public Pixmap {};
00091   typedef Bitmap IconImage;
00092 }
00093
00094 #endif //GDCMICONIMAGE_H

```

13.309 gdcmIconImageFilter.h File Reference

```

#include "gdcmFile.h"
#include "gdcmIconImage.h"

```



```

00053 ~IconImageFilter();
00054
00056 void SetFile(const File& f) { F = f; }
00057 File &GetFile() { return *F; }
00058 const File &GetFile() const { return *F; }
00059
00061 bool Extract();
00062
00064 unsigned int GetNumberOfIconImages() const;
00065 IconImage& GetIconImage( unsigned int i ) const;
00066
00067 protected:
00068 void ExtractIconImages();
00069 void ExtractVeprolIconImages();
00070
00071 private:
00072 SmartPointer<File> F;
00073 IconImageFilterInternals *Internals;
00074 };
00075
00076 } // end namespace gdcm
00077
00078 #endif //GDCMICONIMAGEFILTER_H

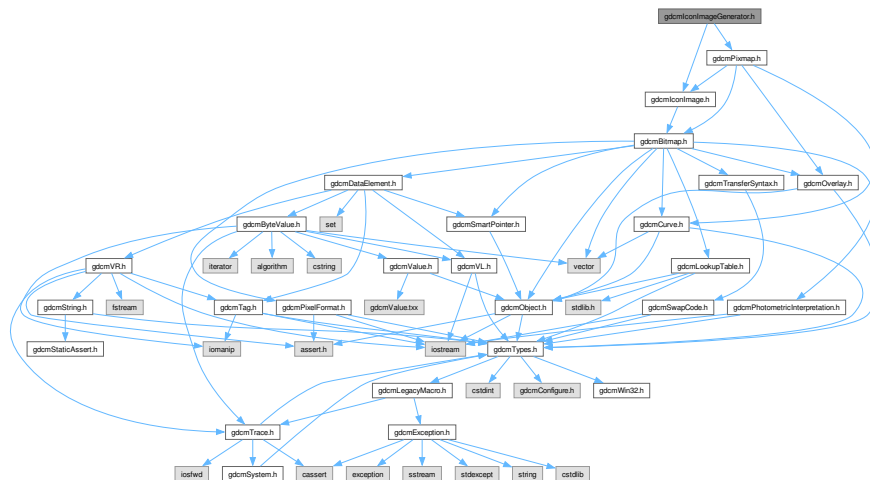
```

13.311 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
```

```
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class [gdcm::IconImageGenerator](#)
[IconImageGenerator](#).

Namespaces

- namespace [gdcm](#)

13.312 gdcmIconImageGenerator.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMICONIMAGEGENERATOR_H
00015 #define GDCMICONIMAGEGENERATOR_H
00016
00017 #include "gdcmPixmap.h"
00018 #include "gdcmIconImage.h"
00019
00020 namespace gdcm
00021 {
00022     class IconImageGeneratorInternals;
00041     class GDCM_EXPORT IconImageGenerator
00042     {
00043     public:
00044         IconImageGenerator();
00045         ~IconImageGenerator();
00046
00048         void SetPixmap(const Pixmap& p) { P = p; }
00049         Pixmap &GetPixmap() { return *P; }
00050         const Pixmap &GetPixmap() const { return *P; }
00051
00053         void SetOutputDimensions(const unsigned int dims[2]);
00054
00058         void SetPixelMinMax(double min, double max);
00059
00063         void AutoPixelMinMax(bool b);
00064
00069         void ConvertRGBToPaletteColor(bool b);
00070
00074         void SetOutsideValuePixel(double v);
00075
00077         bool Generate();
00078
00080         const IconImage& GetIconImage() const { return *I; }
00081
00082     protected:
00083
00084     private:
00085         void BuildLUT( Bitmap & bitmap, unsigned int maxcolor );
00086
00087         SmartPointer<Pixmap> P;
00088         SmartPointer<IconImage> I;
00089         IconImageGeneratorInternals *Internals;
00090     };
00091
00092 } // end namespace gdcm
00093
00094 #endif //GDCMICONIMAGEGENERATOR_H

```

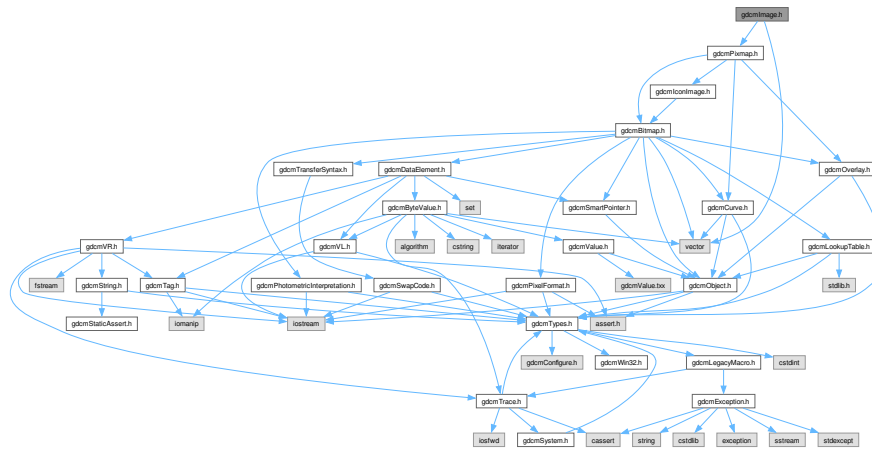
13.313 gdcmImage.h File Reference

```

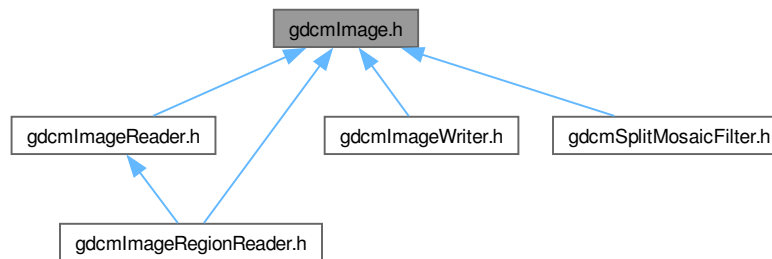
#include "gdcmPixmap.h"
#include <vector>

```


Include dependency graph for gdcImage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::Image`
Image.

Namespaces

- namespace `gdcm`

13.314 gdcmImage.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMAGE_H
00015 #define GDCMIMAGE_H
00016
00017 #include "gdcmPixmap.h"
00018
00019 #include <vector>
00020
00021 namespace gdcm
00022 {
00023
00046 class GDCM_EXPORT Image : public Pixmap
00047 {
00048 public:
00049   Image():Spacing(),SC(),Intercept(0),Slope(1) {
00050     //DirectionCosines.resize(6);
00051     Origin.resize( 3 /*NumberOfDimensions*/ ); // fill with 0
00052     DirectionCosines.resize( 6 ); // fill with 0
00053     DirectionCosines[0] = 1;
00054     DirectionCosines[4] = 1;
00055     Spacing.resize( 3 /*NumberOfDimensions*/, 1 ); // fill with 1
00056   }
00057   ~Image() override = default;
00058
00063   const double *GetSpacing() const;
00064   double GetSpacing(unsigned int idx) const;
00065   void SetSpacing(const double spacing[3]);
00066   void SetSpacing(unsigned int idx, double spacing);
00067
00070   const double *GetOrigin() const;
00071   double GetOrigin(unsigned int idx) const;
00072   void SetOrigin(const float origin[3]);
00073   void SetOrigin(const double origin[3]);
00074   void SetOrigin(unsigned int idx, double ori);
00075
00078   const double *GetDirectionCosines() const;
00079   double GetDirectionCosines(unsigned int idx) const;
00080   void SetDirectionCosines(const float dircos[6]);
00081   void SetDirectionCosines(const double dircos[6]);
00082   void SetDirectionCosines(unsigned int idx, double dircos);
00083
00085   void Print(std::ostream &os) const override;
00086
00088   void SetIntercept(double intercept) { Intercept = intercept; }
00089   double GetIntercept() const { return Intercept; }
00090
00092   void SetSlope(double slope) { Slope = slope; }
00093   double GetSlope() const { return Slope; }
00094
00095 private:
00096   std::vector<double> Spacing;
00097   std::vector<double> Origin;
00098   std::vector<double> DirectionCosines;
00099
00100   // I believe the following 3 ivars can be derived from TS ...
00101   SwapCode SC;
00102   double Intercept;
00103   double Slope;
00104 };
00105

```

13.315 gdcmImageApplyLookupTable.h File Reference

Include dependency graph for `gdcmImageApplyLookupTable.h`:



- ## Namespaces

- 13.316 gdcImageApplyLookupTable.h

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even

```

```

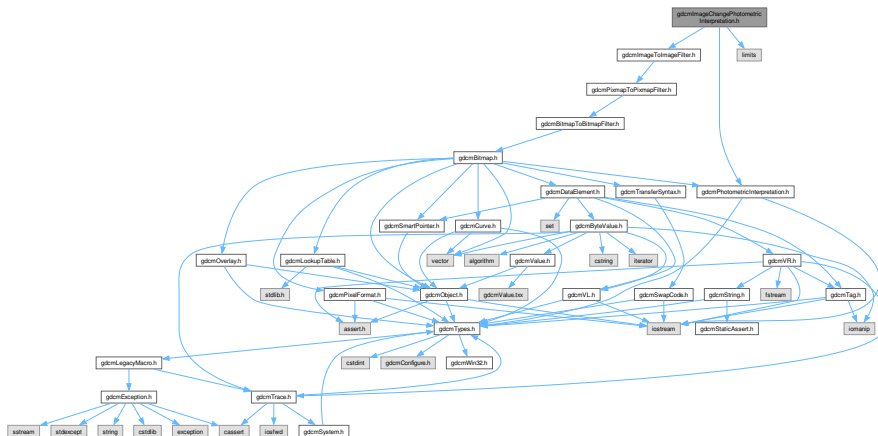
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
=====*/
00014 #ifndef GDCMIMAGEAPPLYLOOKUPTABLE_H
00015 #define GDCMIMAGEAPPLYLOOKUPTABLE_H
00016
00017 #include "gdcmImageToImageFilter.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class DataElement;
00023 class GDCM_EXPORT ImageApplyLookupTable : public ImageToImageFilter
00024 {
00025 public:
00026     ImageApplyLookupTable();
00027     ~ImageApplyLookupTable();
00028
00029     bool Apply();
00030
00031     void SetRGB8(bool b);
00032
00033 protected:
00034
00035 private:
00036     struct impl;
00037     // PIMPL idiom
00038     impl* pimpl;
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMIMAGEAPPLYLOOKUPTABLE_H

```

13.317 gdcImageChangePhotometricInterpretation.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
#include <limits>
```

Include dependency graph for `gdcImageChangePhotometricInterpretation.h`:



Classes

- class `gdcm::ImageChangePhotometricInterpretation`
`ImageChangePhotometricInterpretation` class.

Namespaces

- namespace [gdcm](#)

Functions

- [template<typename T> static T gdcm::Clamp \(int v\)](#)
- [template<typename T> static int gdcm::Round \(T x\)](#)

13.318 gdcmImageChangePhotometricInterpretation.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00015  #define GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00016
00017  #include "gdcmImageToImageFilter.h"
00018  #include "gdcmPhotometricInterpretation.h"
00019  #include <limits>
00020
00021  namespace gdcm
00022  {
00023
00024  class DataElement;
00029  class GDCM_EXPORT ImageChangePhotometricInterpretation : public ImageToImageFilter
00030  {
00031  public:
00032  ImageChangePhotometricInterpretation():PI() {}
00033  ~ImageChangePhotometricInterpretation() = default;
00034
00036  void SetPhotometricInterpretation(PhotometricInterpretation const &pi) { PI = pi; }
00037  const PhotometricInterpretation &GetPhotometricInterpretation() const { return PI; }
00038
00040  bool Change();
00041
00044  template <typename T>
00045  static void RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits = 8);
00046  template <typename T>
00047  static void YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits = 8);
00048
00049  protected:
00050  bool ChangeMonochrome();
00051  bool ChangeYBR2RGB();
00052  bool ChangeRGB2YBR();
00053
00054  private:
00055  PhotometricInterpretation PI;
00056  };
00057
00058  template <typename T>
00059  static inline int Round(T x)
00060  {

```

```

00061     return (int)(x+0.5);
00062 }
00063
00064 template <typename T>
00065 static inline T Clamp(int v)
00066 {
00067     gdcmm_assert( std::numeric_limits<T>::min() == 0 );
00068     return v < 0 ? 0 : (v > std::numeric_limits<T>::max() ? std::numeric_limits<T>::max() : v);
00069 }
00070
00071
00072 template <typename T>
00073 void ImageChangePhotometricInterpretation::RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits)
00074 {
00075     // Implementation details, since the equations from:
00076     // http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.3.html#sect_C.7.6.3.1.2
00077     // are rounded to the 4th decimal precision, prefer the exact equation from the original document at:
00078     // CCIR Recommendation 601-2, also found in T.871 (Section §7, page 4)
00079     const double R = rgb[0];
00080     const double G = rgb[1];
00081     const double B = rgb[2];
00082     gdcmm_assert( storedbits <= sizeof(T) * 8 );
00083     const int halffullscale = 1 « (storedbits - 1);
00084     const int Y = Round( 0.299 * R + 0.587 * G + 0.114 * B );
00085     const int CB = Round((-0.299 * R - 0.587 * G + 0.886 * B)/1.772 + halffullscale);
00086     const int CR = Round(( 0.701 * R - 0.587 * G - 0.114 * B)/1.402 + halffullscale);
00087     ybr[0] = Clamp<T>(Y);
00088     ybr[1] = Clamp<T>(CB);
00089     ybr[2] = Clamp<T>(CR);
00090 }
00091
00092 template <typename T>
00093 void ImageChangePhotometricInterpretation::YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits)
00094 {
00095     const double Y = ybr[0];
00096     const double Cb = ybr[1];
00097     const double Cr = ybr[2];
00098     gdcmm_assert( storedbits <= sizeof(T) * 8 );
00099     const int halffullscale = 1 « (storedbits - 1);
00100     const int R = Round(Y + 1.402 * (Cr-halffullscale) );
00101     const int G = Round(Y -( 0.114 * 1.772 * (Cb-halffullscale) + 0.299 * 1.402 * (Cr-halffullscale))/0.587);
00102     const int B = Round(Y + 1.772 * (Cb-halffullscale) );
00103     rgb[0] = Clamp<T>(R);
00104     rgb[1] = Clamp<T>(G);
00105     rgb[2] = Clamp<T>(B);
00106 }
00107
00108 } // end namespace gdcmm
00109
00110 #endif //GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H

```

```
#include "gdcmImageToImageFilter.h"
```

- class `gdcm::ImageChangePlanarConfiguration`
`ImageChangePlanarConfiguration` class.

- namespace `gdcm`

[Go to the documentation of this file.](#)

Generated by Doxygen

```

00018
00019 namespace gdcmm
00020 {
00021
00022 class DataElement;
00023 class GDCM_EXPORT ImageChangePlanarConfiguration : public ImageToImageFilter
00024 {
00025 public:
00026 ImageChangePlanarConfiguration():PlanarConfiguration(0) {}
00027 ~ImageChangePlanarConfiguration() = default;
00028
00029 void SetPlanarConfiguration(unsigned int pc) { PlanarConfiguration = pc; }
00030 unsigned int GetPlanarConfiguration() const { return PlanarConfiguration; }
00031
00032 template <typename T>
00033 static size_t RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s);
00034
00035 template <typename T>
00036 static size_t RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T *rgb, size_t s);
00037
00038 bool Change();
00039
00040 protected:
00041
00042 private:
00043 unsigned int PlanarConfiguration;
00044 };
00045
00046 template <typename T>
00047 size_t ImageChangePlanarConfiguration::RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s)
00048 {
00049 T *pout = out;
00050 for(size_t i = 0; i < s; ++i )
00051 {
00052 *pout++ = *r++;
00053 *pout++ = *g++;
00054 *pout++ = *b++;
00055 }
00056
00057 gdcmm_assert( (size_t)(pout - out) == 3 * s );
00058 return pout - out;
00059 }
00060
00061 template <typename T>
00062 size_t ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T *rgb, size_t s)
00063 {
00064 const T *prgb = rgb;
00065 for(size_t i = 0; i < s; ++i )
00066 {
00067 *r++ = *prgb++;
00068 *g++ = *prgb++;
00069 *b++ = *prgb++;
00070 }
00071 gdcmm_assert( (size_t)(prgb - rgb) == 3 * s );
00072 return prgb - rgb;
00073 }
00074
00075 } // end namespace gdcmm
00076
00077 #endif //GDCMIMAGECHANGEPLANARCONFIGURATION_H

```

13.321 gdcmmImageChangeTransferSyntax.h File Reference

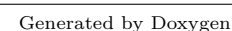
```

#include "gdcmmImageToImageFilter.h"
#include "gdcmmTransferSyntax.h"

```


13.323 gdcImageCodec.h File Reference

Include dependency graph for gdcmImageCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::ImageCodec`
`ImageCodec`.

Namespaces

- namespace `gdcm`

13.324 gdcmImageCodec.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMAGECODEC_H
00015 #define GDCMIMAGECODEC_H
00016
00017 #include "gdcmCodec.h"
00018 #include "gdcmPhotometricInterpretation.h"
00019 #include "gdcmLookupTable.h"
00020 #include "gdcmSmartPointer.h"
00021 #include "gdcmPixelFormat.h"
00022
00023 namespace gdcm
00024 {
00025
00030 class GDCM_EXPORT ImageCodec : public Codec
00031 {
00032     friend class ImageChangePhotometricInterpretation;
00033 public:
00034     ImageCodec();
00035     ~ImageCodec() override;
00036     bool CanCode(TransferSyntax const &) const override { return false; }
00037     bool CanDecode(TransferSyntax const &) const override { return false; }
00038     bool Decode(DataElement const &is_, DataElement &os) override;
00039     bool IsLossy() const;
00040     void SetLossyFlag(bool l);
00041     bool GetLossyFlag() const;
00042
00043     virtual bool GetHeaderInfo(std::istream &is_, TransferSyntax &ts);
00044
00045     virtual ImageCodec * Clone() const = 0;

```

```

00046
00047 protected:
00048     bool DecodeByStreams(std::istream &is_, std::ostream &os) override;
00049     virtual bool IsValid(PhotometricInterpretation const &pi);
00050 public:
00051
00052     unsigned int GetPlanarConfiguration() const
00053     {
00054         return PlanarConfiguration;
00055     }
00056     void SetPlanarConfiguration(unsigned int pc)
00057     {
00058         gdcmm_assert( pc == 0 || pc == 1 );
00059         PlanarConfiguration = pc;
00060     }
00061
00062     PixelFormat &GetPixelFormat()
00063     {
00064         return PF;
00065     }
00066     const PixelFormat &GetPixelFormat() const
00067     {
00068         return PF;
00069     }
00070     virtual void SetPixelFormat(PixelFormat const &pf)
00071     {
00072         PF = pf;
00073     }
00074     const PhotometricInterpretation &GetPhotometricInterpretation() const;
00075     void SetPhotometricInterpretation(PhotometricInterpretation const &pi);
00076
00077     bool GetNeedByteSwap() const
00078     {
00079         return NeedByteSwap;
00080     }
00081     void SetNeedByteSwap(bool b)
00082     {
00083         NeedByteSwap = b;
00084     }
00085     void SetNeedOverlayCleanup(bool b)
00086     {
00087         NeedOverlayCleanup = b;
00088     }
00089     void SetLUT(LookupTable const &lut)
00090     {
00091         LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
00092     }
00093     const LookupTable &GetLUT() const
00094     {
00095         return *LUT;
00096     }
00097
00098     void SetDimensions(const unsigned int d[3]);
00099     void SetDimensions(const std::vector<unsigned int> &d);
00100     const unsigned int *GetDimensions() const { return Dimensions; }
00101     void SetNumberOfDimensions(unsigned int dim);
00102     unsigned int GetNumberOfDimensions() const;
00103
00104     bool CleanupUnusedBits(char * data, size_t datalen);
00105
00106 protected:
00107     // Streaming (write) API:
00114     friend class FileChangeTransferSyntax;
00115     virtual bool StartEncode( std::ostream & os );
00116     virtual bool IsRowEncoder();
00117     virtual bool IsFrameEncoder();
00118     virtual bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen );
00119     virtual bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen );
00120     virtual bool StopEncode( std::ostream & os);
00121
00122 protected:
00123     bool RequestPlanarConfiguration;
00124     bool RequestPaddedCompositePixelCode;
00125     //private:
00126     unsigned int PlanarConfiguration;
00127     PhotometricInterpretation PI;
00128     PixelFormat PF;
00129     bool NeedByteSwap;
00130     bool NeedOverlayCleanup;
00131
00132     typedef SmartPointer<LookupTable> LUTPtr;

```

```

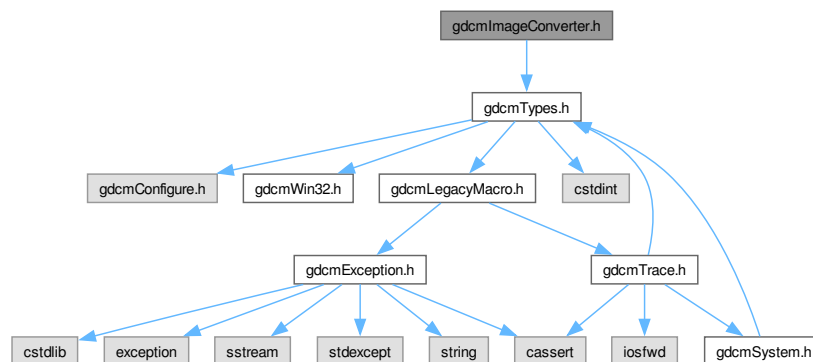
00133  LUTPtr LUT;
00134  unsigned int Dimensions[3]; // FIXME
00135  unsigned int NumberOfDimensions;
00136  bool LossyFlag;
00137
00138  bool DoOverlayCleanup(std::istream &is_, std::ostream &os);
00139  bool DoByteSwap(std::istream &is_, std::ostream &os);
00140  bool DoYBR(std::istream &is_, std::ostream &os);
00141  bool DoYBRFull422(std::istream &is_, std::ostream &os);
00142  bool DoPlanarConfiguration(std::istream &is_, std::ostream &os);
00143  bool DoSimpleCopy(std::istream &is_, std::ostream &os);
00144  bool DoPaddedCompositePixelCode(std::istream &is_, std::ostream &os);
00145  bool DoInvertMonochrome(std::istream &is_, std::ostream &os);
00146
00147  //template <typename T>
00148  //bool DoInvertPlanarConfiguration(T *output, const T *input, uint32_t length);
00149  };
00150
00151 } // end namespace gdcm
00152
00153 #endif //GDCMIMAGECODEC_H

```

13.325 gdcmImageConverter.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmImageConverter.h:



Classes

- class `gdcm::ImageConverter`
Image Converter.

Namespaces

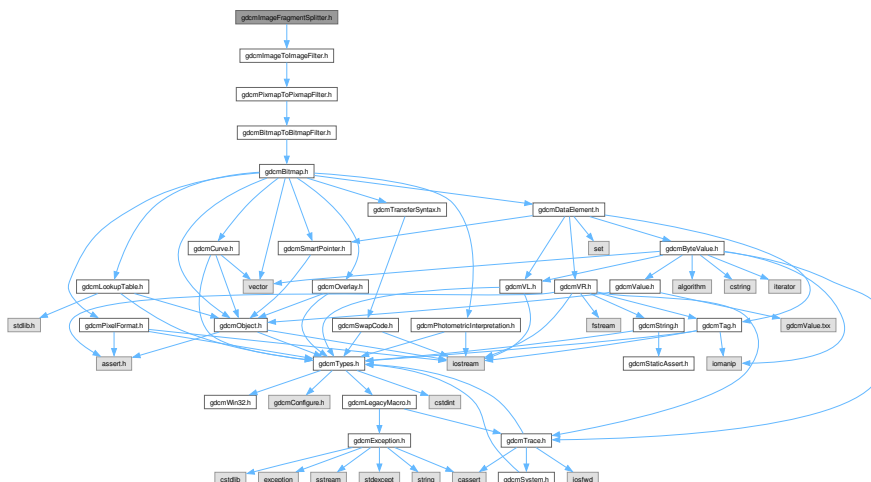
- namespace `gdcm`

13.326 gdcmImageConverter.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  00015  #ifndef GDCMIMAGECONVERTER_H
00016  #define GDCMIMAGECONVERTER_H
00017  00018  #include "gdcmTypes.h"
00019  00020  namespace gdcm
00021  {
00022  00023  class Image;
00024  00025  class GDCM__EXPORT ImageConverter
00026  {
00027  public:
00028  00029  ImageConverter();
00030  ~ImageConverter();
00031  00032  void SetInput(Image const &input);
00033  const Image& GetOutput() const;
00034  00035  void Convert();
00036  private:
00037  Image *Input;
00038  Image *Output;
00039  };
00040  } // end namespace gdcm
00041  00042  #endif //GDCMIMAGECONVERTER_H
```

Include dependency graph for `gdcmImageFragmentSplitter.h`:



- class `gdcm::ImageFragmentSplitter`
`ImageFragmentSplitter` class.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```

00001  /*=====*/
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGEFRAGMENTSPLITTER_H
00015  #define GDCMIMAGEFRAGMENTSPLITTER_H
00016
00017  #include "gdcmImageToImageFilter.h"

```

```

00018
00019 namespace gdcm
00020 {
00021
00022 class DataElement;
00027 class GDCM_EXPORT ImageFragmentSplitter : public ImageToImageFilter
00028 {
00029 public:
00030 ImageFragmentSplitter():FragmentSizeMax(0),Force(false) {}
00031 ~ImageFragmentSplitter() = default;
00032
00034 bool Split();
00035
00037 void SetFragmentSizeMax(unsigned int fragsize);
00038 unsigned int GetFragmentSizeMax() const { return FragmentSizeMax; }
00039
00042 void SetForce( bool f ) { Force = f; }
00043
00044 protected:
00045
00046 private:
00047 unsigned int FragmentSizeMax;
00048 bool Force;
00049 };
00050
00051 } // end namespace gdcm
00052
00053 #endif //GDCMIMAGEFRAGMENTSPLITTER_H

```

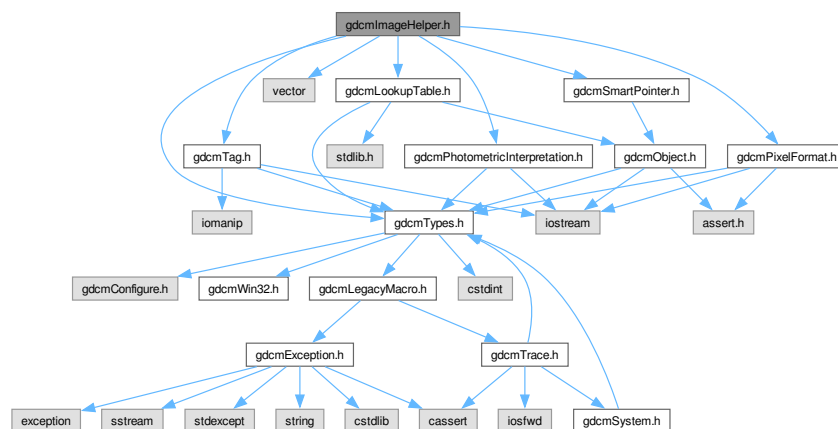
13.329 gdcmImageHelper.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"

```

Include dependency graph for gdcmImageHelper.h:



Classes

- class [gdcm::ImageHelper](#)

[ImageHelper](#) (internal class, not intended for user level).

- struct [gdcm::RealWorldValueMappingContent](#)

Namespaces

- namespace [gdcm](#)

13.330 gdcmImageHelper.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMAGEHELPER_H
00015 #define GDCMIMAGEHELPER_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTag.h"
00019 #include <vector>
00020 #include "gdcmPixelFormat.h"
00021 #include "gdcmPhotometricInterpretation.h"
00022 #include "gdcmSmartPointer.h"
00023 #include "gdcmLookupTable.h"
00024
00025 namespace gdcm
00026 {
00027
00028 class MediaStorage;
00029 class DataSet;
00030 class File;
00031 class Image;
00032 class Pixmap;
00033 class ByteValue;
00034
00035 // minimal struct:
00036 struct RealWorldValueMappingContent {
00037     double RealWorldValueIntercept;
00038     double RealWorldValueSlope;
00039     // http://dicom.nema.org/MEDICAL/DICOM/2014c/output/chtml/part16/sect_CID_7181.html
00040     std::string CodeValue;
00041     std::string CodeMeaning;
00042 };
00043
00044
00045 class GDCM_EXPORT ImageHelper
00046 {
00047 public:
00048     static void SetForceRescaleInterceptSlope(bool);
00049     static bool GetForceRescaleInterceptSlope();
00050
00051     static void SetPMSRescaleInterceptSlope(bool);
00052     static bool GetPMSRescaleInterceptSlope();
00053
00054     static void SetForcePixelSpacing(bool);
00055     static bool GetForcePixelSpacing();
00056
00057     static void SetSecondaryCaptureImagePlaneModule(bool);
00058     static bool GetSecondaryCaptureImagePlaneModule();
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
00073
00074
00075
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094

```

```

00098 static std::vector<unsigned int> GetDimensionsValue(const File& f);
00099 static void SetDimensionsValue(File& f, const Pixmap & img);
00100
00103 static PixelFormat GetPixelFormatValue(const File& f);
00104
00109 static std::vector<double> GetRescaleInterceptSlopeValue(File const & f);
00110 static void SetRescaleInterceptSlopeValue(File & f, const Image & img);
00111
00112 // read only for now
00113 static bool GetRealWorldValueMappingContent(File const & f, RealWorldValueMappingContent & rwvmc);
00114
00116 static std::vector<double> GetOriginValue(File const & f);
00117 static void SetOriginValue(DataSet & ds, const Image & img);
00118
00121 static std::vector<double> GetDirectionCosinesValue(File const & f);
00127 // FIXME: There is a major issue for image with multiple IOP (eg. Enhanced * Image Storage).
00128 static void SetDirectionCosinesValue(DataSet & ds, const std::vector<double> & dircos);
00129
00131 static std::vector<double> GetSpacingValue(File const & f);
00133 static void SetSpacingValue(DataSet & ds, const std::vector<double> & spacing);
00134
00136 static bool ComputeSpacingFromImagePositionPatient(const std::vector<double> &imageposition, std::vector<double> &
spacing);
00137
00138 static bool GetDirectionCosinesFromDataSet(DataSet const & ds, std::vector<double> & dircos);
00139
00140 //functions to get more information from a file
00141 //useful for the stream image reader, which fills in necessary image information
00142 //distinctly from the reader-style data input
00143 static PhotometricInterpretation GetPhotometricInterpretationValue(File const& f);
00144 //returns the configuration of colors in a plane, either RGB RGB RGB or RRR GGG BBB
00145 static unsigned int GetPlanarConfigurationValue(const File& f);
00146
00148 static SmartPointer<LookupTable> GetLUT(File const& f);
00149
00150 // Moved from PixampReader to here. Generally used for photometric interpretation.
00151 static const ByteValue* GetPointerFromElement(Tag const &tag, File const& f);
00152
00154 static MediaStorage ComputeMediaStorageFromModality(const char *modality,
00155 unsigned int dimension = 2, PixelFormat const & pf = PixelFormat(),
00156 PhotometricInterpretation const & pi = PhotometricInterpretation(),
00157 double rescaleintercept = 0, double rescaleslope = 1 );
00158
00159 protected:
00160 static Tag GetSpacingTagFromMediaStorage(MediaStorage const &ms);
00161 static Tag GetZSpacingTagFromMediaStorage(MediaStorage const &ms);
00162
00163 private:
00164 static bool ForceRescaleInterceptSlope;
00165 static bool PMSRescaleInterceptSlope;
00166 static bool ForcePixelSpacing;
00167 static bool SecondaryCaptureImagePlaneModule;
00168 };
00169
00170 } // end namespace gdcm
00171
00172 #endif // GDCMIMAGEHELPER_H

```

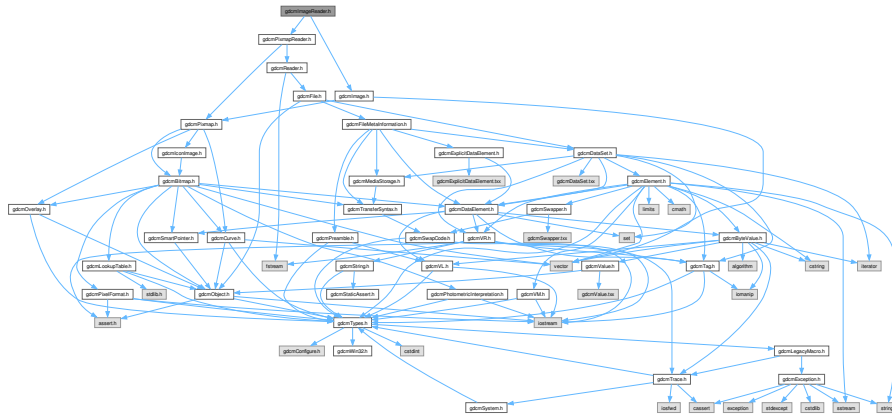
13.331 gdcmImageReader.h File Reference

```

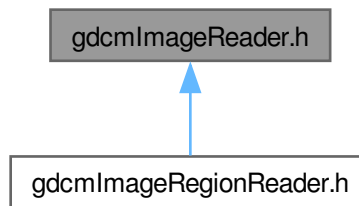
#include "gdcmPixmapReader.h"
#include "gdcmImage.h"

```

Include dependency graph for gdcmImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::ImageReader`
`ImageReader`.

Namespaces

- namespace `gdcm`

13.332 gdcImageReader.h

Go to the documentation of this file.

00001

```

00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMIMAGEREADER_H
00015 #define GDCMIMAGEREADER_H
00016
00017 #include "gdcmPixmapReader.h"
00018 #include "gdcmImage.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class MediaStorage;
00024 class GDCM_EXPORT ImageReader : public PixmapReader
00025 {
00026 public:
00027 ImageReader();
00028 ~ImageReader() override;//needs to be virtual to ensure lack of memory leaks
00029
00030 bool Read() override;
00031
00032 // Following methods are valid only after a call to 'Read'
00033 const Image& GetImage() const;
00034 Image& GetImage();
00035 //void SetImage(Image const &img);
00036
00037 protected:
00038 bool ReadImage(MediaStorage const &ms) override;
00039 bool ReadACRNEMAImage() override;
00040 };
00041 } // end namespace gdcm
00042
00043 #endif //GDCMIMAGEREADER_H

```

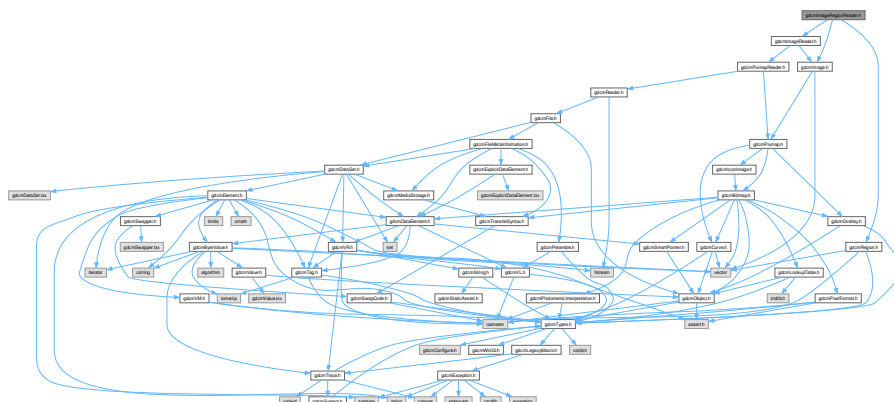
13.333 gdcmImageRegionReader.h File Reference

```

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"

```

Include dependency graph for gdcmImageRegionReader.h:



Classes

- class `gdcm::ImageRegionReader`
`ImageRegionReader`.

Namespaces

- namespace `gdcm`

13.334 gdcmImageRegionReader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGEEXTENTREADER_H
00015  #define GDCMIMAGEEXTENTREADER_H
00016
00017  #include "gdcmImageReader.h"
00018  #include "gdcmImage.h"
00019  #include "gdcmRegion.h"
00020
00021  namespace gdcm
00022  {
00023
00024  class ImageRegionReaderInternals;
00034  class GDCM_EXPORT ImageRegionReader : public ImageReader
00035  {
00036  public:

```

```

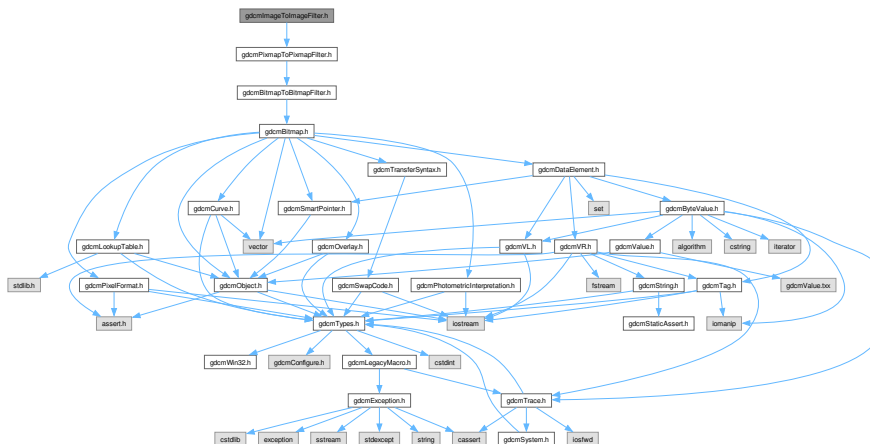
00037 ImageRegionReader();
00038 ~ImageRegionReader() override;
00039
00041 void SetRegion(Region const & region);
00042 Region const &GetRegion() const;
00043
00047 size_t ComputeBufferLength() const;
00048
00051 bool ReadInformation();
00052
00057 bool ReadIntoBuffer(char *inreadbuffer, size_t buflen);
00058
00059 protected:
00061     bool Read() override;
00062
00063 private:
00064     BoxRegion ComputeBoundingBox();
00065     bool ReadRAWIntoBuffer(char *buffer, size_t buflen);
00066     bool ReadRLEIntoBuffer(char *buffer, size_t buflen);
00067     bool ReadJPEG2000IntoBuffer(char *buffer, size_t buflen);
00068     bool ReadJPEGIntoBuffer(char *buffer, size_t buflen);
00069     bool ReadJPEGLSIntoBuffer(char *buffer, size_t buflen);
00070     ImageRegionReaderInternals *Internals;
00071 };
00072
00073 } // end namespace gdcim
00074
00075 #endif //GDCMIMAGEEXTENTREADER_H

```

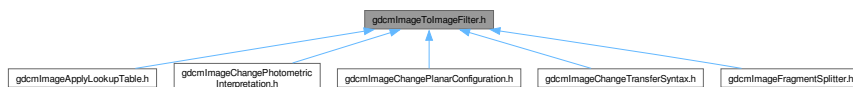
13.335 gdcImageToImageFilter.h File Reference

```
#include "gdcmPidmapToPīmapFilter.h"
```

Include dependency graph for `gdcmImageToImageFilter.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ImageToImageFilter](#)
[ImageToImageFilter](#) class.

Namespaces

- namespace [gdcm](#)

13.336 gdcmImageToImageFilter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMIMAGETOIMAGEFILTER_H
00015  #define GDCMIMAGETOIMAGEFILTER_H
00016
00017  #include "gdcmPixmapToPixmapFilter.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class Image;
00027  class GDCM_EXPORT ImageToImageFilter : public PixmapToPixmapFilter
00028  {
00029  public:
00030  ImageToImageFilter();
00031  ~ImageToImageFilter() = default;
00032
00033  Image &GetInput();
00034
00035  // NOTE: covariant return-type to preserve backward compatible API
00037  const Image &GetOutput() const;
00038
00039  protected:
00040  };
00041
00042  } // end namespace gdcm
00043
00044  #endif //GDCMIMAGETOIMAGEFILTER_H

```

13.337 gdcmImageWriter.h File Reference

```

#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"

```



```

00034 public:
00035     ImageWriter();
00036     ~ImageWriter() override;
00037
00041     const Image& GetImage() const override { return dynamic_cast<const Image*>(*PixelData); }
00042     Image& GetImage() override { return dynamic_cast<Image*>(*PixelData); } // FIXME
00043     //void SetImage(Image const &img);
00044
00046     bool Write() override; // Execute()
00047
00050     MediaStorage ComputeTargetMediaStorage();
00051 protected:
00052
00053 private:
00054 };
00055
00056 } // end namespace gdcm
00057
00058 #endif //GDCMIMAGEWRITER_H

```

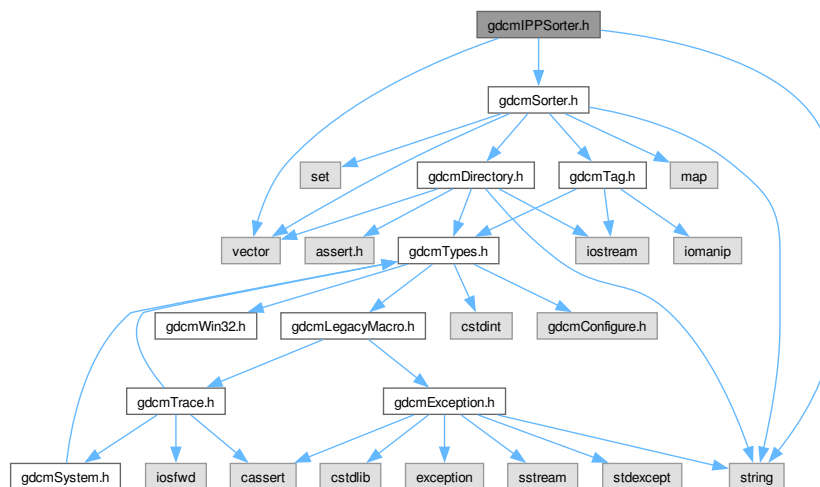
13.339 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
```

```
#include <vector>
```

```
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class [gdcm::IPPSorter](#)
IPPSorter.

Namespaces

- namespace [gdcm](#)

13.340 gdcmIPPSorter.h

[Go to the documentation of this file.](#)

```

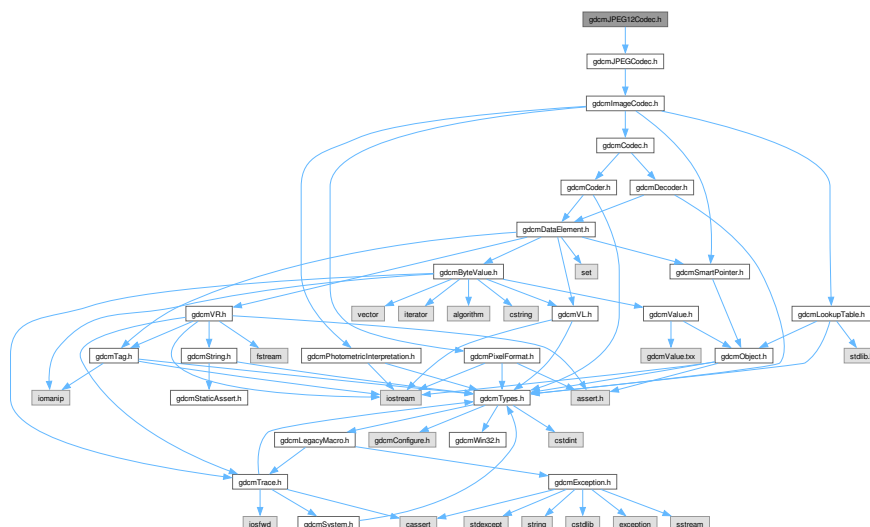
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIPPSORTER_H
00015 #define GDCMIPPSORTER_H
00016
00017 #include "gdcmSorter.h"
00018
00019 #include <vector>
00020 #include <string>
00021
00022 namespace gdcm
00023 {
00043 class GDCM__EXPORT IPPSorter : public Sorter
00044 {
00045 public:
00046   IPPSorter();
00047
00048   // FIXME: I do not like public virtual function...
00055   bool Sort(std::vector<std::string> const & filenames) override;
00056
00064   void SetComputeZSpacing(bool b) { ComputeZSpacing = b; }
00068   void SetZSpacingTolerance(double tol) { ZTolerance = tol; }
00069   double GetZSpacingTolerance() const { return ZTolerance; }
00070
00080   void SetDirectionCosinesTolerance(double tol) { DirCosTolerance = tol; }
00081   double GetDirectionCosinesTolerance() const { return DirCosTolerance; }
00082
00086   void SetDropDuplicatePositions(bool b) { DropDuplicatePositions = b; }
00087
00094   double GetZSpacing() const { return ZSpacing; }
00095
00096 protected:
00097   bool ComputeZSpacing;
00098   bool DropDuplicatePositions;
00099   double ZSpacing;
00100   double ZTolerance;
00101   double DirCosTolerance;
00102
00103 private:
00104   GDCM__LEGACY(bool ComputeSpacing(std::vector<std::string> const & filenames))
00105 };
00106
00107
00108 } // end namespace gdcm
00109
00110 #endif //GDCMIPPSORTER_H

```

13.341 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class [gdcm::JPEG12Codec](#)
Class to do JPEG 12bits (lossy & lossless).

Namespaces

- namespace [gdcm](#)

13.342 gdcmJPEG12Codec.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMJPEG12CODEC_H
00015 #define GDCMJPEG12CODEC_H
```


Namespaces

- namespace [gdcm](#)

13.344 gdcmJPEG16Codec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMJPEG16CODEC_H
00015  #define GDCMJPEG16CODEC_H
00016  00017  #include "gdcmJPEGCodec.h"
00018  00019  namespace gdcm
00020  {
00021  00022  class JPEGInternals_16BIT;
00023  class ByteValue;
00028  class JPEG16Codec : public JPEGCodec
00029  {
00030  public:
00031  JPEG16Codec();
00032  ~JPEG16Codec() override;
00033  00034  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00035  bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
00036  00037  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00038  00039  protected:
00040  bool IsStateSuspension() const override;
00041  bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
00042  00043  private:
00044  JPEGInternals_16BIT *Internals;
00045  };
00046  00047  } // end namespace gdcm
00048  00049  #endif //GDCMJPEG16CODEC_H

```



```

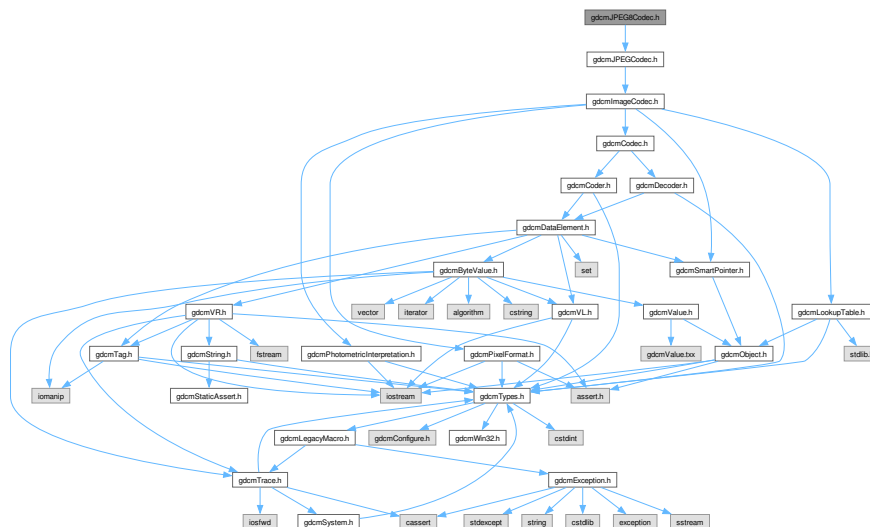
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class JPEG2000Internals;
00030 class GDCM_EXPORT JPEG2000Codec : public ImageCodec
00031 {
00032 friend class ImageRegionReader;
00033 friend class Bitmap;
00034 public:
00035 JPEG2000Codec();
00036 ~JPEG2000Codec() override;
00037
00038 bool CanDecode(TransferSyntax const &ts) const override;
00039 bool CanCode(TransferSyntax const &ts) const override;
00040
00041 bool Decode(DataElement const &is, DataElement &os) override;
00042 bool Code(DataElement const &in, DataElement &out) override;
00043
00044 bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00045 ImageCodec * Clone() const override;
00046
00047 // JPEG-2000 / OpenJPEG specific way of encoding lossy-ness
00048 // ref: http://www.openjpeg.org/index.php?menu=doc#encoder
00049 void SetRate(unsigned int idx, double rate);
00050 double GetRate(unsigned int idx = 0) const;
00051
00052 void SetQuality(unsigned int idx, double q);
00053 double GetQuality(unsigned int idx = 0) const;
00054
00055 void SetTileSize(unsigned int tx, unsigned int ty);
00056
00057 void SetNumberOfResolutions(unsigned int nres);
00058
00061 void SetNumberOfThreadsForDecompression(int nThreads);
00062
00063 void SetReversible(bool res);
00064 void SetMCT(unsigned int mct);
00065
00066 protected:
00067 bool DecodeExtent(
00068     char *buffer,
00069     unsigned int xmin, unsigned int xmax,
00070     unsigned int ymin, unsigned int ymax,
00071     unsigned int zmin, unsigned int zmax,
00072     std::istream & is
00073 );
00074
00075 bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00076
00077 bool StartEncode( std::ostream & ) override;
00078 bool IsRowEncoder() override;
00079 bool IsFrameEncoder() override;
00080 bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00081 bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00082 bool StopEncode( std::ostream & ) override;
00083
00084 private:
00085 std::pair<char *, size_t> DecodeByStreamsCommon(char *dummy_buffer, size_t buf_size);
00086 bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t &complen, const char * indata, size_t inlen );
00087 bool GetHeaderInfo(const char * dummy_buffer, size_t len, TransferSyntax &ts);
00088 JPEG2000Internals *Internals;
00089 };
00090
00091 } // end namespace gdcm
00092
00093 #endif //GDCMJPEG2000CODEC_H

```

13.347 gdcmJPEG8Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG8Codec.h:



Classes

- class [gdcm::JPEG8Codec](#)
Class to do JPEG 8bits (lossy & lossless).

Namespaces

- namespace [gdcm](#)

13.348 gdcmJPEG8Codec.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMJPEG8CODEC_H
00015 #define GDCMJPEG8CODEC_H
```



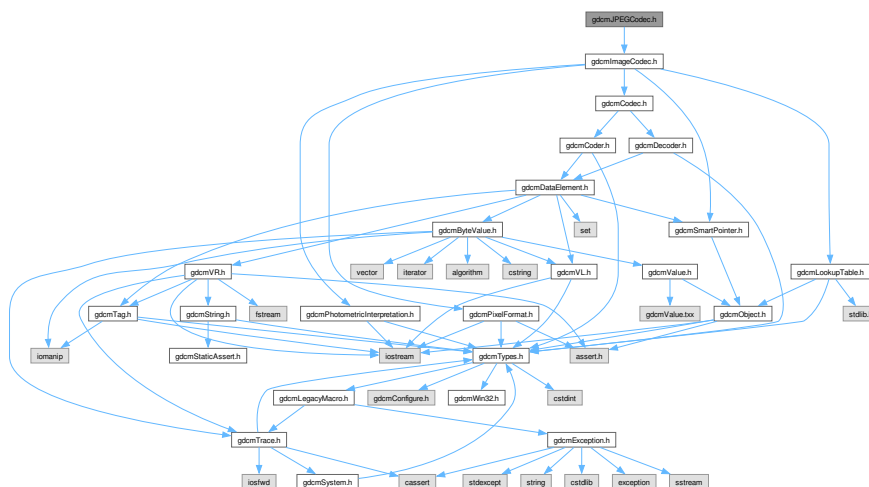
```

00016
00017 #include "gdcm.JPEGCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class JPEGInternals_8BIT;
00023 class ByteValue;
00028 class JPEG8Codec : public JPEGCodec
00029 {
00030 public:
00031     JPEG8Codec();
00032     ~JPEG8Codec() override;
00033
00034     bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00035     bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;
00036
00037     bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00038
00039 protected:
00040     bool IsStateSuspension() const override;
00041     bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;
00042
00043 private:
00044     JPEGInternals_8BIT *Internals;
00045 };
00046
00047 } // end namespace gdcm
00048
00049 #endif //GDCMJPEG8CODEC H

```

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::JPEGCodec](#)
JPEG codec.

Namespaces

- namespace [gdcm](#)

13.350 gdcMJPEGCodec.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMJPEGCODEC_H
00015  #define GDCMJPEGCODEC_H
00016
00017  #include "gdcmImageCodec.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class PixelFormat;
00023  class TransferSyntax;
00040  class GDCM__EXPORT JPEGCodec : public ImageCodec
00041  {
00042  friend class ImageRegionReader;
00043  public:
00044  JPEGCodec();
00045  ~JPEGCodec() override;
00046  bool CanDecode(TransferSyntax const &ts) const override;
00047  bool CanCode(TransferSyntax const &ts) const override;
00048  bool Decode(DataElement const &is, DataElement &os) override;

```

```

00049 void SetPixelFormat(PixelFormat const &pf) override;
00050
00052 void ComputeOffsetTable(bool b);
00053
00055 bool Code(DataElement const &in, DataElement &out) override;
00056
00057 bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00058 ImageCodec * Clone() const override;
00059
00060 //void SetReversible(bool res);
00061
00062 void SetQuality(double q);
00063 double GetQuality() const;
00064
00065 void SetLossless(bool l);
00066 bool GetLossless() const;
00067
00068 virtual bool EncodeBuffer( std::ostream & out,
00069     const char *inbuffer, size_t inlen);
00070
00071 protected:
00072 bool DecodeExtent(
00073     char *buffer,
00074     unsigned int xmin, unsigned int xmax,
00075     unsigned int ymin, unsigned int ymax,
00076     unsigned int zmin, unsigned int zmax,
00077     std::istream & is
00078 );
00079
00080 bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00081 bool IsValid(PhotometricInterpretation const &pi) override;
00082
00083 bool StartEncode( std::ostream & ) override;
00084 bool IsRowEncoder() override;
00085 bool IsFrameEncoder() override;
00086 bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00087 bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00088 bool StopEncode( std::ostream & ) override;
00089
00090 protected:
00091 // Internal method called by SetPixelFormat
00092 // Instantiate the right jpeg codec (8, 12 or 16)
00093 void SetBitSample(int bit);
00094
00095 virtual bool IsStateSuspension() const;
00096
00097 protected:
00098 int BitSample;
00099 //bool Lossless;
00100 int Quality;
00101
00102 private:
00103 void SetupJPEGBitCodec(int bit);
00104 JPEGCodec *Internal;
00105 };
00106
00107 } // end namespace gdcm
00108
00109 #endif //GDCMJPEGCODEC_H

```



```

00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class JPEGLSInternals;
00030 class GDCM__EXPORT JPEGLSCodec : public ImageCodec
00031 {
00032 friend class ImageRegionReader;
00033 public:
00034   JPEGLSCodec();
00035   ~JPEGLSCodec() override;
00036   bool CanDecode(TransferSyntax const &ts) const override;
00037   bool CanCode(TransferSyntax const &ts) const override;
00038
00039   unsigned long GetBufferLength() const { return BufferLength; }
00040   void SetBufferLength(unsigned long l) { BufferLength = l; }
00041
00042   bool Decode(DataElement const &is, DataElement &os) override;
00043   bool Decode(DataElement const &in, char* outBuffer, size_t inBufferLength,
00044             uint32_t inXMin, uint32_t inXMax, uint32_t inYMin,
00045             uint32_t inYMax, uint32_t inZMin, uint32_t inZMax);
00046   bool Code(DataElement const &in, DataElement &out) override;
00047
00048   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00049   ImageCodec * Clone() const override;
00050
00051   void SetLossless(bool l);
00052   bool GetLossless() const;
00053
00054   /*
00055    * test.acr can look pretty bad, even with a lossy error of 2. Explanation follows:
00056    * I agree that the test image looks ugly. In this particular case I can
00057    * explain though.
00058    *
00059    * The image is 8 bit, but it does not use the full 8 bit dynamic range. The
00060    * black pixels have value 234 and the white 255. If you set allowed lossy
00061    * error to 2, you allow an error of about 10% of the actual dynamic range.
00062    * That is of course very visible.
00063    */
00065   void SetLossyError(int error);
00066
00067 protected:
00068   bool DecodeExtent(
00069     char *buffer,
00070     unsigned int xmin, unsigned int xmax,
00071     unsigned int ymin, unsigned int ymax,
00072     unsigned int zmin, unsigned int zmax,
00073     std::istream & is
00074   );
00075
00076   bool StartEncode( std::ostream & ) override;
00077   bool IsRowEncoder() override;
00078   bool IsFrameEncoder() override;
00079   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00080   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00081   bool StopEncode( std::ostream & ) override;
00082
00083 private:
00084   bool DecodeByStreamsCommon(const char *buffer, size_t totalLen, std::vector<unsigned char> &rgbyteOut);
00085   bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t inlen );
00086
00087   unsigned long BufferLength;
00088   int LossyError;
00089 };
00090
00091 } // end namespace gdcm
00092
00093 #endif //GDCMJPEGLSCODEC_H

```

13.353 gdcmJSON.h File Reference

```

#include "gdcmFile.h"
#include "gdcmDataElement.h"

```

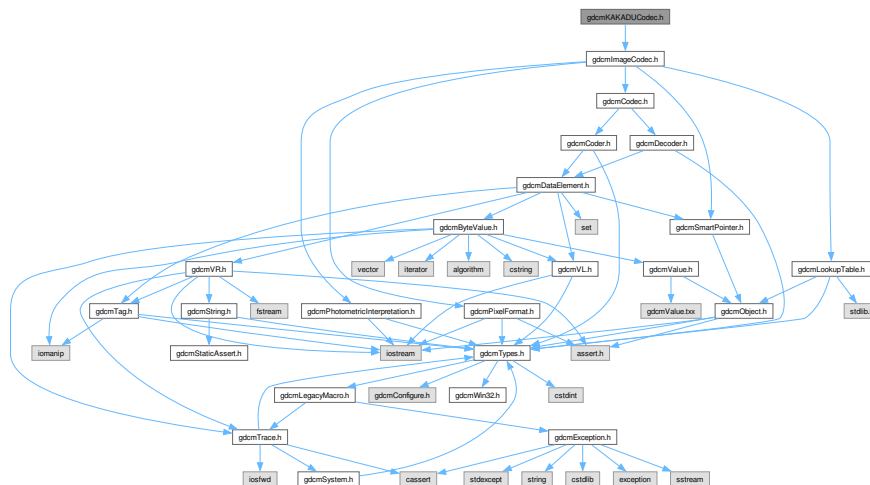


```

00026 {
00027
00028     class JSONInternal;
00029     class GDCM_EXPORT JSON
00030     {
00031     public:
00032         JSON();
00033         ~JSON();
00034
00035         bool GetPrettyPrint() const;
00036         void SetPrettyPrint(bool onoff);
00037         void PrettyPrintOn();
00038         void PrettyPrintOff();
00039
00040         bool Code(DataSet const & in, std::ostream & os);
00041         bool Decode(std::istream & is, DataSet & out);
00042
00043     private:
00044         JSONInternal *Internals;
00045     };
00046
00047 } // end namespace gdcmm
00048
00049 #endif //GDCMXMLPRINTER_H

```

```
#include "gdcmImageCodec.h"
Include dependency graph for gdcmKAKADUCodec.h:
```



- class `gdcm::KAKADUCodec`
`KAKADUCodec`.

- namespace `gdcm`

13.356 gdcmKAKADUCodec.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMKAKADUCODEC_H
00015 #define GDCMKAKADUCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class KAKADUCodec : public ImageCodec
00023 {
00024 public:
00025     KAKADUCodec();
00026     ~KAKADUCodec() override;
00027     bool CanDecode(TransferSyntax const &ts) const override;
00028     bool CanCode(TransferSyntax const &ts) const override;
00029
00030     bool Decode(DataElement const &is, DataElement &os) override;
00031     bool Code(DataElement const &in, DataElement &out) override;
00032
00033     ImageCodec * Clone() const override;
00034 private:
00035 };
00036
00037 } // end namespace gdcm
00038
00039 #endif //GDCMKAKADUCODEC_H

```

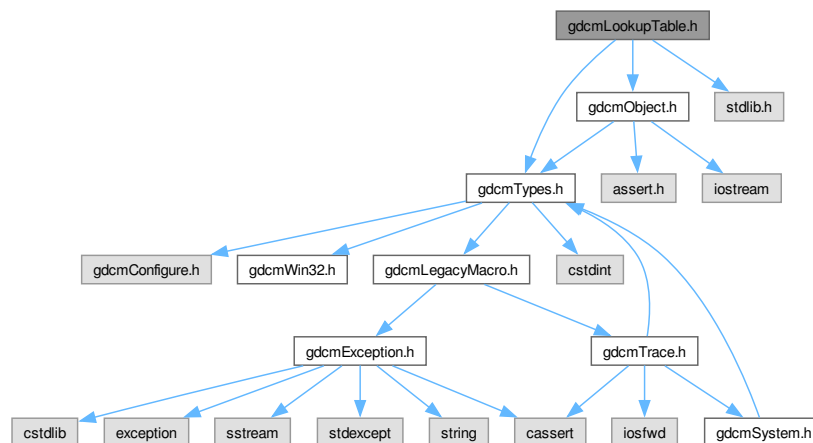
13.357 gdcmLookupTable.h File Reference

```

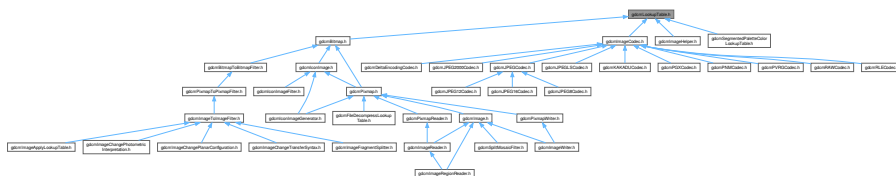
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>

```


Include dependency graph for gdcmlLookupTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::LookupTable](#)
LookupTable class.

Namespaces

- namespace [gdcml](#)

13.358 gdcmlLookupTable.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
  
```

```

00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014
00015 #ifndef GDCMLOOKUPTABLE_H
00016 #define GDCMLOOKUPTABLE_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmObject.h"
00020 #include <stdlib.h>
00021
00022 namespace gdcm
00023 {
00024
00025 class LookupTableInternal;
00029 class GDCM_EXPORT LookupTable : public Object
00030 {
00031 public:
00032     typedef enum {
00033         RED = 0, // Keep RED == 0
00034         GREEN,
00035         BLUE,
00036         GRAY,
00037         UNKNOWN
00038     } LookupTableType;
00039
00040     LookupTable();
00041     ~LookupTable() override;
00042     void Print(std::ostream &) const override;
00043
00045     void Allocate( unsigned short bitsample = 8 );
00047     //TODO: check to see if length should be unsigned short, unsigned int, or whatever
00048     void InitializeLUT(LookupTableType type, unsigned short length,
00049         unsigned short subscript, unsigned short bitsize);
00050     unsigned int GetLUTLength(LookupTableType type) const;
00051     virtual void SetLUT(LookupTableType type, const unsigned char *array,
00052         unsigned int length);
00053     void GetLUT(LookupTableType type, unsigned char *array, unsigned int &length) const;
00054     void GetLUTDescriptor(LookupTableType type, unsigned short &length,
00055         unsigned short &subscript, unsigned short &bitsize) const;
00056
00058     void InitializeRedLUT(unsigned short length, unsigned short subscript,
00059         unsigned short bitsize);
00060     void SetRedLUT(const unsigned char *red, unsigned int length);
00061     void InitializeGreenLUT(unsigned short length, unsigned short subscript,
00062         unsigned short bitsize);
00063     void SetGreenLUT(const unsigned char *green, unsigned int length);
00064     void InitializeBlueLUT(unsigned short length, unsigned short subscript,
00065         unsigned short bitsize);
00066     void SetBlueLUT(const unsigned char *blue, unsigned int length);
00067
00069     void Clear();
00070
00072     void Decode(std::istream &is, std::ostream &os) const;
00073
00077     bool Decode(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
00078
00080     bool IsRGB8() const;
00081
00083     bool Decode8(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;
00084
00085     LookupTable(LookupTable const &lut):Object(lut), Internal(nullptr), BitSample(0), IncompleteLUT(false)
00086     {
00087         gdcm_assert(0);
00088     }
00089
00091     bool GetBufferAsRGBA(unsigned char *rgba) const;
00092
00094     const unsigned char *GetPointer() const;
00095
00097     bool WriteBufferAsRGBA(const unsigned char *rgba);
00098
00100     unsigned short GetBitSample() const { return BitSample; }
00101
00103     bool Initialized() const;

```

```

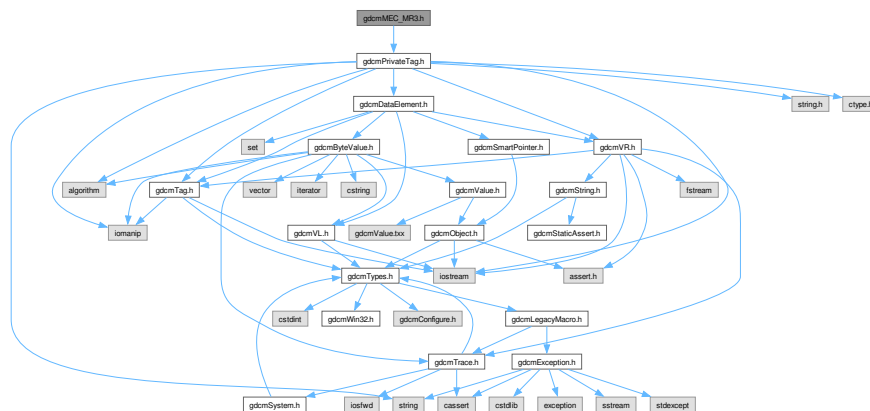
00104
00105 private:
00107 void Encode(std::istream &is, std::ostream &os);
00108
00109 protected:
00110 LookupTableInternal *Internal;
00111 unsigned short BitSample; // refer to the pixel type (not the bit size of LUT)
00112 bool IncompleteLUT:1;
00113 };
00114
00115 } // end namespace gdcm
00116
00117 #endif //GDCMLOOKUPTABLE_H

```

13.359 gdcmMEC_MR3.h File Reference

#include "gdcmPrivateTag.h"

Include dependency graph for gdcmMEC_MR3.h:



Classes

- class [gdcm::MEC_MR3](#)
Class for [MEC_MR3](#).

Namespaces

- namespace [gdcm](#)

13.360 gdcmMEC_MR3.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library

```

```

00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMMEC_MR3_H
00015 #define GDCMMEC_MR3_H
00016
00017 #include "gdcmPrivateTag.h"
00018
00019 namespace gdcm {
00024 class GDCM_EXPORT MEC_MR3 {
00025 public:
00026 static bool Print(const char *src, size_t srclen);
00027
00030 static const PrivateTag &GetPMTFInformationDataTag();
00031
00034 static const PrivateTag &GetCanonMECMR3Tag();
00035
00038 static const PrivateTag &GetToshibaMECMR3Tag();
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif // GDCMMEC_MR3_H

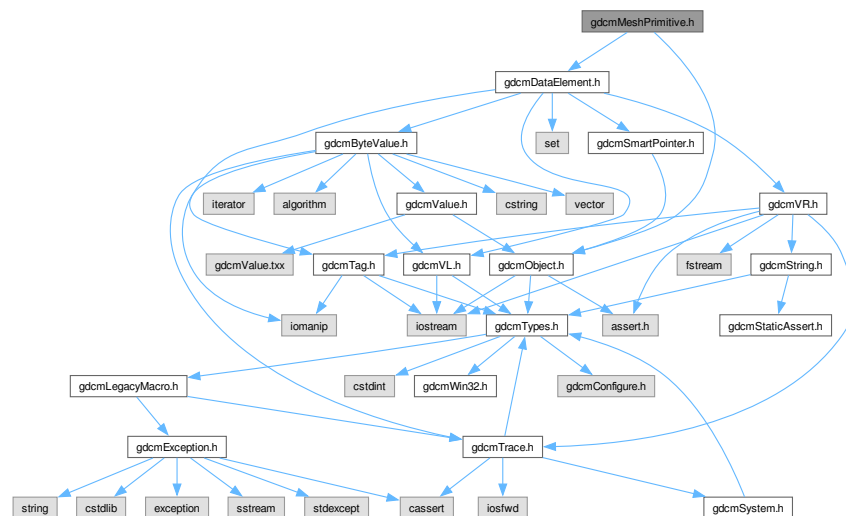
```

13.361 gdcmMeshPrimitive.h File Reference

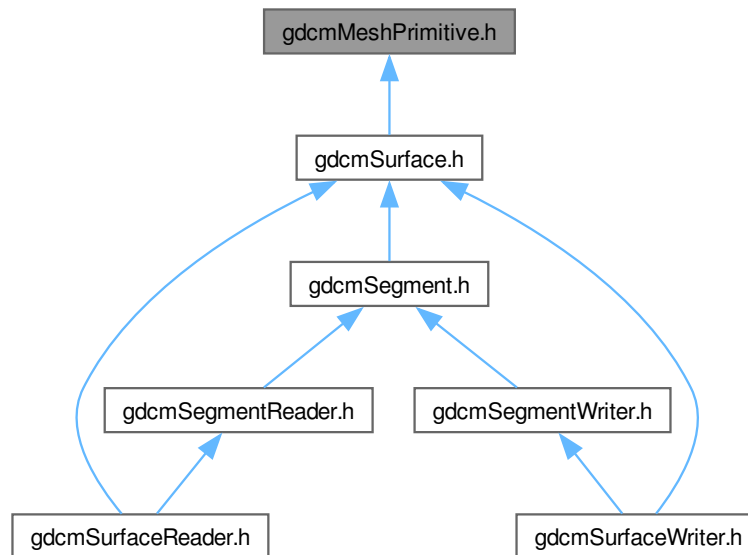
```
#include <gdcmObject.h>
```

```
#include <gdcmDataElement.h>
```

Include dependency graph for gdcmMeshPrimitive.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::MeshPrimitive`

This class defines surface mesh primitives.

Namespaces

- namespace `gdcm`

13.362 gdcmmeshprimitive.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*

```

```

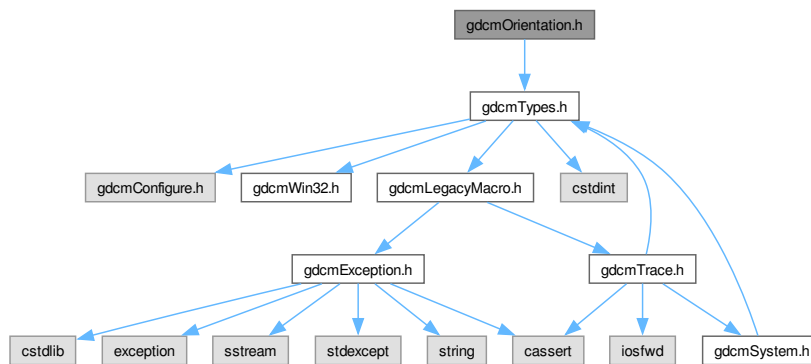
00014
00015 #ifndef GDCMMESHPRIMITIVE_H
00016 #define GDCMMESHPRIMITIVE_H
00017
00018 #include <gdcmObject.h>
00019 #include <gdcmDataElement.h>
00020
00021 namespace gdcm
00022 {
00023
00024 class GDCM_EXPORT MeshPrimitive : public Object
00025 {
00026 public:
00027
00028 typedef std::vector< DataElement > PrimitivesData;
00029
00030 typedef enum {
00031     VERTEX = 0,
00032     EDGE,
00033     TRIANGLE,
00034     TRIANGLE_STRIP,
00035     TRIANGLE_FAN,
00036     LINE,
00037     FACET,
00038     MPType_END
00039 } MPType;
00040
00041 static const char * GetMPTypeString(const MPType type);
00042
00043 static MPType GetMPType(const char * type);
00044
00045 MeshPrimitive();
00046
00047 ~MeshPrimitive() override;
00048
00049 MPType GetPrimitiveType() const;
00050 void SetPrimitiveType(const MPType type);
00051
00052 const DataElement & GetPrimitiveData() const;
00053 DataElement & GetPrimitiveData();
00054 void SetPrimitiveData(DataElement const & de);
00055
00056 const PrimitivesData & GetPrimitivesData() const;
00057 PrimitivesData & GetPrimitivesData();
00058 void SetPrimitivesData(PrimitivesData const & DEs);
00059
00060 const DataElement & GetPrimitiveData(const unsigned int idx) const;
00061 DataElement & GetPrimitiveData(const unsigned int idx);
00062 void SetPrimitiveData(const unsigned int idx, DataElement const & de);
00063 void AddPrimitiveData(DataElement const & de);
00064
00065 unsigned int GetNumberOfPrimitivesData() const;
00066
00067 protected:
00068
00069 // Use to define tag where PrimitiveData will be put.
00070 MPType PrimitiveType;
00071
00072 // PrimitiveData contains point index list.
00073 // It shall have 1 or 1-n DataElement following PrimitiveType.
00074 PrimitivesData PrimitiveData;
00075 };
00076
00077 }
00078
00079 #endif // GDCMMESHPRIMITIVE_H

```

13.363 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



Classes

- class [gdcm::Orientation](#)
class to handle [Orientation](#)

Namespaces

- namespace [gdcm](#)

Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const [Orientation](#) &o)

13.364 gdcmOrientation.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012

```

```

00013
00014 =====*/
00014 #ifndef GDCMORIENTATION_H
00015 #define GDCMORIENTATION_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class GDCM_EXPORT Orientation
00023 {
00024     friend std::ostream& operator<<(std::ostream &_os, const Orientation &o);
00025 public:
00026     Orientation();
00027     ~Orientation() = default;
00028
00029     void Print(std::ostream &) const;
00030
00031     typedef enum {
00032         UNKNOWN,
00033         AXIAL,
00034         CORONAL,
00035         SAGITTAL,
00036         OBLIQUE
00037     } OrientationType;
00038
00039     static OrientationType GetType(const double dircos[6]);
00040
00041     static void SetObliquityThresholdCosineValue(double val);
00042     static double GetObliquityThresholdCosineValue();
00043
00044     static const char *GetLabel(OrientationType type);
00045
00046 protected:
00047     static char GetMajorAxisFromPatientRelativeDirectionCosine(double x, double y, double z);
00048
00049 private:
00050     static double ObliquityThresholdCosineValue;
00051 };
00052 //-----
00053 inline std::ostream& operator<<(std::ostream &os, const Orientation &o)
00054 {
00055     o.Print(os);
00056     return os;
00057 }
00058
00059 } // end namespace gdcm
00060
00061 #endif //GDCMORIENTATION_H

```

13.365 gdcmOverlay.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmObject.h"

```



```

graph TD
    gdcmOverlay.h --> gdcmObject.h
    gdcmOverlay.h --> gdcmTypes.h
    gdcmObject.h --> gdcmTypes.h
    gdcmObject.h --> assert.h
    gdcmObject.h --> iostream
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> cstdint
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmTrace.h --> cassert
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h
    gdcmTypes.h --> gdcmSystem.h
  
```

[illegible]

- class `gdcm::Overlay`
`Overlay` class.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```

00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMOVERLAY_H
00015 #define GDCMOVERLAY_H
00016
00017 #include "gdcmtypes.h"
00018 #include "gdcmlObject.h"
00019
00020 namespace gdcml
00021 {
00022
00023 class OverlayInternal;
00024 class ByteValue;
00025 class DataSet;
00026 class DataElement;
00027 class GDCM_EXPORT Overlay : public Object
00028 {
00029 public:
00030 Overlay();
00031 ~Overlay() override;
00032 void Print(std::ostream &) const override;
00033
00034 void Update(const DataElement & de);
00035
00036 void SetGroup(unsigned short group);
00037 unsigned short GetGroup() const;
00038 void SetRows(unsigned short rows);
00039 unsigned short GetRows() const;
00040 void SetColumns(unsigned short columns);
00041 unsigned short GetColumns() const;
00042 void SetNumberOfFrames(unsigned int numberofframes);
00043 void SetDescription(const char* description);
00044 const char* GetDescription() const;
00045 typedef enum {
00046 Invalid = 0,
00047 Graphics = 1,
00048 ROI = 2
00049 } OverlayType;
00050 void SetType(const char* type);
00051 const char* GetType() const;
00052 OverlayType GetTypeAsEnum() const;
00053 static const char* GetOverlayTypeAsString(OverlayType ot);
00054 static OverlayType GetOverlayTypeFromString(const char*);
00055 void SetOrigin(const signed short origin[2]);
00056 const signed short* GetOrigin() const;
00057 void SetFrameOrigin(unsigned short frameorigin);
00058 void SetBitsAllocated(unsigned short bitsallocated);
00059 unsigned short GetBitsAllocated() const;
00060 void SetBitPosition(unsigned short bitposition);
00061 unsigned short GetBitPosition() const;
00062
00063 void SetOverlay(const char* array, size_t length);
00064
00065 bool GrabOverlayFromPixelData(DataSet const &ds);
00066
00067 const ByteValue &GetOverlayData() const;
00068
00069 bool IsEmpty() const;
00070
00071 bool IsZero() const;
00072
00073 bool IsInPixelData() const;
00074 void IsInPixelData(bool b);
00075
00076 void Decompress(std::ostream &os) const;
00077
00078 size_t GetUnpackBufferLength() const;
00079 bool GetUnpackBuffer(char* buffer, size_t len) const;

```

```

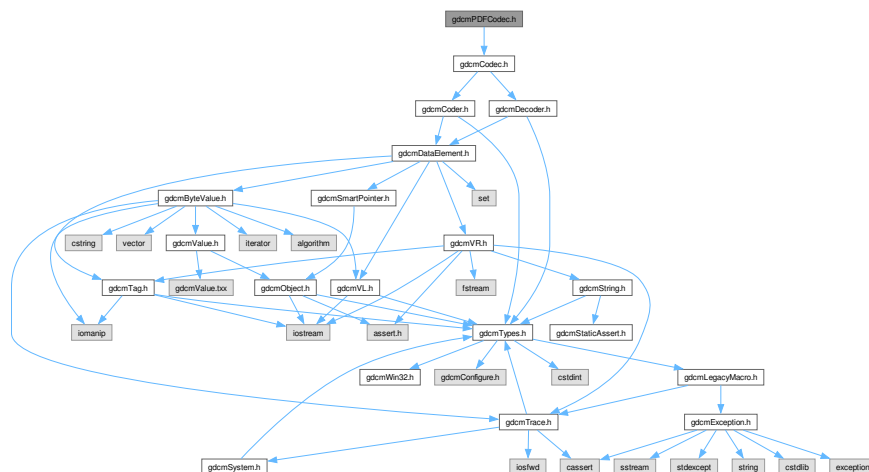
00130
00131  Overlay(Overlay const &ov);
00132  Overlay &operator=(Overlay const &ov);
00133
00134 private:
00135  OverlayInternal *Internal;
00136 };
00137
00138 } // end namespace gdcm
00139
00140 #endif //GDCM_OVERLAY_H

```

13.367 gdcmPDFCodec.h File Reference

#include "gdcmCodec.h"

Include dependency graph for gdcmPDFCodec.h:



Classes

- class [gdcm::PDFCodec](#)
PDFCodec class.

Namespaces

- namespace [gdcm](#)

13.368 gdcmPDFCodec.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====

```

```

00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMPDFCODEC_H
00015 #define GDCMPDFCODEC_H
00016
00017 #include "gdcmlib.h"
00018
00019 namespace gdcmlib
00020 {
00021
00022 class GDCM_EXPORT PDFCodec : public Codec
00023 {
00024 public:
00025 PDFCodec();
00026 ~PDFCodec() override;
00027 bool CanCode(TransferSyntax const &) const override { return false; }
00028 bool CanDecode(TransferSyntax const &) const override { return false; }
00029 bool Decode(DataElement const &is, DataElement &os) override;
00030 };
00031
00032 } // end namespace gdcmlib
00033
00034 #endif //GDCMPDFCODEC_H

```

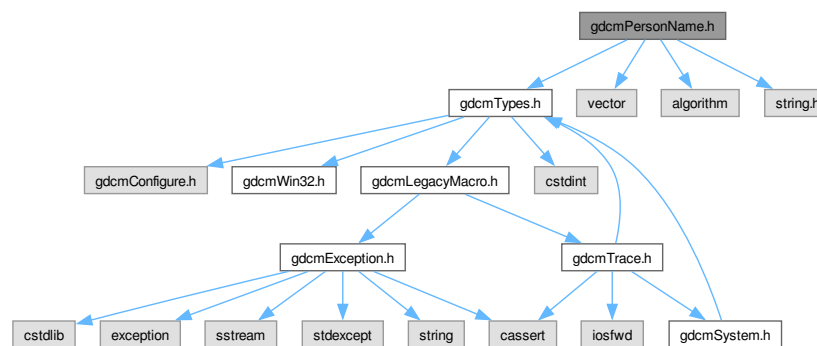
13.369 gdcmlibPersonName.h File Reference

```

#include "gdcmlibTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>

```

Include dependency graph for gdcmlibPersonName.h:



Classes

- class `gdcmlib::PersonName`
PersonName class.

Namespaces

- namespace [gdcm](#)

13.370 gdcmPersonName.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015 #ifndef GDCMPERSONNAME_H
00016 #define GDCMPERSONNAME_H
00017
00018 #include "gdcmTypes.h"
00019 #include <vector>
00020 #include <algorithm> // std::min
00021 #include <string.h> // strlen
00022
00023 namespace gdcm
00024 {
00025
00026 class GDCM_EXPORT PersonName
00027 {
00028 public:
00029 static const unsigned int MaxNumberOfComponents = 5;
00030 static const unsigned int MaxLength = 64;
00031 char Component[MaxNumberOfComponents][MaxLength+1];
00032 static const char Separator = ',';
00033 static const char Padding = ' ';
00034
00035 unsigned int GetNumberOfComponents() const {
00036 unsigned int r = 0;
00037 for(unsigned int i = 0; i < 5; ++i) {
00038 if( *Component[i] != '\0' ) r = i;
00039 }
00040 return r+1;
00041 }
00042 unsigned int GetMaxLength() const { return MaxLength; }
00043 void SetBlob(const std::vector<char>& v) {
00044 (void)v;
00045 //gdcm_assert(0); //TODO
00046 }
00047 void SetComponents(const char *comp1 = "",
00048 const char *comp2 = "",
00049 const char *comp3 = "",
00050 const char *comp4 = "",
00051 const char *comp5 = "") {
00052 const char *components[5] = { comp1, comp2, comp3, comp4, comp5 };
00053 SetComponents( components );
00054 }
00055 void SetComponents(const char *components[]) {
00056 if( components )
00057 for(unsigned int i = 0; i < 5; ++i) {
00058 if( components[i] && strlen(components[i]) < GetMaxLength() )
00059 strcpy(Component[i], components[i]);
00060 gdcm_assert( strlen(Component[i]) < GetMaxLength() );
00061 }
00062 }
00063 void Print(std::ostream &os) const
00064 {
00065 //os << "Family Name Complex: " << Component[0] << std::endl;

```

```

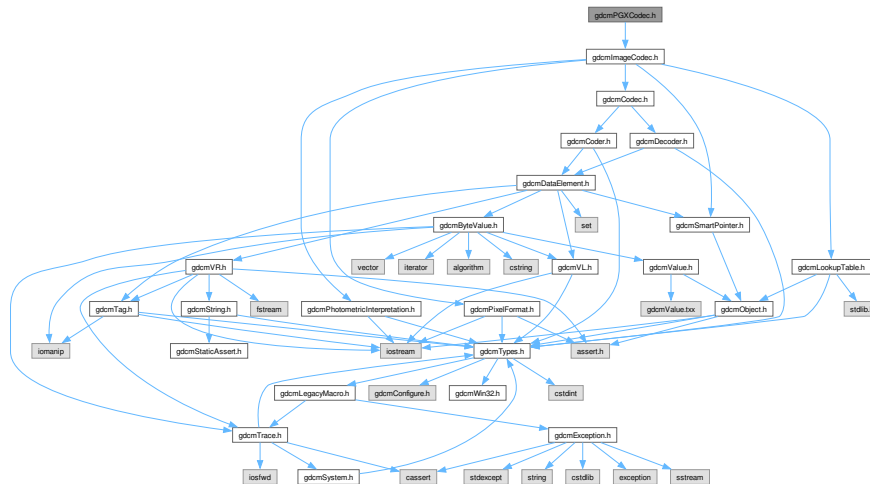
00069 //os « "Given Name Complex: " « Component[1] « std::endl;
00070 //os « "Middle Name      : " « Component[2] « std::endl;
00071 //os « "Name Suffix      : " « Component[3] « std::endl;
00072 //os « "Name Prefix      : " « Component[4] « std::endl;
00073 os « Component[0] «
00074 os « Component[1] «
00075 os « Component[2] «
00076 os « Component[3] «
00077 os « Component[4];
00078 }
00079 };
00080
00081 } // end namespace gdcmm
00082
00083 #endif //GDCMPERSONNAME_H

```

13.371 gdcmmPGXCodec.h File Reference

#include "gdcmmImageCodec.h"

Include dependency graph for gdcmmPGXCodec.h:



Classes

- class [gdcmm::PGXCodec](#)
Class to do PGX.

Namespaces

- namespace [gdcmm](#)

13.372 gdcmPGXCodec.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPGXCODEC_H
00015 #define GDCMPGXCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022     class GDCM_EXPORT PGXCodec : public ImageCodec
00023     {
00024     public:
00025         PGXCodec();
00026         ~PGXCodec() override;
00027         bool CanDecode(TransferSyntax const &ts) const override;
00028         bool CanCode(TransferSyntax const &ts) const override;
00029
00030         bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00031         ImageCodec * Clone() const override;
00032
00033         bool Read(const char *filename, DataElement &out) const;
00034         bool Write(const char *filename, const DataElement &out) const;
00035     private:
00036     };
00037
00038 } // end namespace gdcm
00039
00040 #endif //GDCMPGXCODEC_H

```

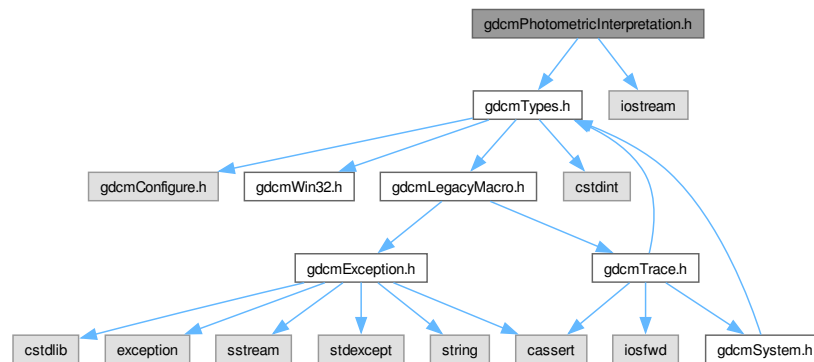
13.373 gdcmPhotometricInterpretation.h File Reference

```

#include "gdcmTypes.h"
#include <iostream>

```

Include dependency graph for `gdcMPhotometricInterpretation.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcM::PhotometricInterpretation](#)
Class to represent an [PhotometricInterpretation](#).

Namespaces

- namespace [gdcM](#)

Functions

- `std::ostream & gdcM::operator<< (std::ostream &os, const PhotometricInterpretation &val)`

13.374 gdcMPhotometricInterpretation.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  #ifndef GDCMPHOTOMETRICINTERPRETATION_H
00016  #define GDCMPHOTOMETRICINTERPRETATION_H
00017
00018  #include "gdcMTypes.h"
00019  #include <iostream>
00020
00021  namespace gdcM
00022  {
00023
00024  class TransferSyntax;
00028  class GDCM_EXPORT PhotometricInterpretation
00029  {
00030  public:
00031      typedef enum {
00032          UNKNOWN = 0,
00033          MONOCHROME1,
00034          MONOCHROME2,
00035          PALETTE_COLOR,
00036          RGB,
00037          HSV,
00038          ARGB, // retired
00039          CMYK,
00040          YBR_FULL,
00041          YBR_FULL_422,
00042          YBR_PARTIAL_422,
00043          YBR_PARTIAL_420,
00044          YBR_ICT,
00045          YBR_RCT,
00046          // PALETTE_COLOR ?
00047          // MONOCHROME = MONOCHROME1 | MONOCHROME2,
00048          // COLOR = RGB | HSV | ARGB | CMYK | YBR_FULL | YBR_FULL_422 | YBR_PARTIAL_422 |
00049          YBR_PARTIAL_420 | YBR_ICT | YBR_RCT,
00050          PI_END // Helpful for internal implementation
00051      } PType; // PhotometricInterpretationType
00052
00053      PhotometricInterpretation(PType pi = UNKNOWN):PIField(pi) {}
00054
00055      static const char *GetPIString(PType pi);
00056
00057      const char *GetString() const;
00058
00059      // You need to make sure end of string is \0
00060      static PType GetPType(const char *pi);
00061
00062      static bool IsRetired(PType pi);
00063
00064      bool IsLossy() const;
00065      bool IsLossless() const;
00066
00067      unsigned short GetSamplesPerPixel() const;
00068
00069      // TODO
00070      // not all PhotometricInterpretation are allowed for compressed Transfer
00071      // syntax
00072      // static bool IsAllowedForCompressedTS(PType pi);
00073
00074      friend std::ostream& operator<<(std::ostream& os, const PhotometricInterpretation& pi);
00075
00076      operator PType () const { return PIField; }

```

13.375 gdcmPixelFormat.h File Reference

Include dependency graph for gdcmPixelFormat.h:



Classes

- class `gdcm::PixelFormat`
`PixelFormat`.

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)`

13.376 gdcmPixelFormat.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  #ifndef GDCMPIXELFORMAT_H
00016  #define GDCMPIXELFORMAT_H
00017
00018  #include "gdcmTypes.h"
00019  #include <iostream>
00020  #include <assert.h>
00021
00022  namespace gdcm
00023  {
00024
00025  class TransferSyntax;
00026
00045  class GDCM_EXPORT PixelFormat
00046  {
00047  friend class Bitmap;
00048  friend std::ostream& operator<<(std::ostream &_os, const PixelFormat &pf);
00049  public:
00050  // When adding a type please add its dual type (its unsigned counterpart)
00051  typedef enum {
00052      UINT8,
00053      INT8,
00054      UINT12,
00055      INT12,
00056      UINT16,
00057      INT16,
00058      UINT32, // For some DICOM files (RT or SC)
00059      INT32,  // " "
00060      UINT64, // Needed when input is 32bits + intercept/slope (incomplete support)
00061      INT64,  // " "
00062      FLOAT16, // sure why not...
00063      FLOAT32, // good ol' 'float'
00064      FLOAT64, // aka 'double'
00065      SINGLEBIT, // bool / monochrome
00066      UNKNOWN // aka BitsAllocated == 0 && PixelRepresentation == 0

```

```

00067 } ScalarType;
00068
00069 // default ctor:
00070 PixelFormat () : PixelFormat(1, 8, 8, 7, 0) {}
00071
00072 explicit PixelFormat (
00073     unsigned short samplesperpixel,
00074     unsigned short bitsallocated = 8,
00075     unsigned short bitsstored = 8,
00076     unsigned short highbit = 7,
00077     unsigned short pixelrepresentation = 0 ) :
00078     SamplesPerPixel(samplesperpixel),
00079     BitsAllocated(bitsallocated),
00080     BitsStored(bitsstored),
00081     HighBit(highbit),
00082     PixelRepresentation(pixelrepresentation) {}
00083 // helper, for the common case
00084 PixelFormat(ScalarType st);
00085
00086 // For transparency of use
00087 operator ScalarType() const { return GetScalarType(); }
00088
00091 unsigned short GetSamplesPerPixel() const;
00092 void SetSamplesPerPixel(unsigned short spp)
00093 {
00094     gdcMAssertMacro( spp <= 4 );
00095     SamplesPerPixel = spp;
00096     gdcM_assert( SamplesPerPixel == 1 || SamplesPerPixel == 3 || SamplesPerPixel == 4 );
00097 }
00098
00100 unsigned short GetBitsAllocated() const
00101 {
00102     return BitsAllocated;
00103 }
00104 void SetBitsAllocated(unsigned short ba)
00105 {
00106     if( ba )
00107     {
00108         switch( ba )
00109         {
00110             /* some devices (FUJIFILM CR + MONO1) incorrectly set BitsAllocated/BitsStored
00111              * as bitmask instead of value. Do what they mean instead of what they say.
00112              */
00113             case 0xffff: ba = 16; break;
00114             case 0x0fff: ba = 12; break;
00115             case 0x00ff: ba = 8; break;
00116         }
00117         BitsAllocated = ba;
00118         BitsStored = ba;
00119         HighBit = (unsigned short)(ba - 1);
00120     }
00121     else // Make the PixelFormat as UNKNOWN
00122     {
00123         BitsAllocated = 0;
00124         PixelRepresentation = 0;
00125     }
00126 }
00127
00129 unsigned short GetBitsStored() const
00130 {
00131     gdcM_assert( BitsStored <= BitsAllocated );
00132     return BitsStored;
00133 }
00134 void SetBitsStored(unsigned short bs)
00135 {
00136     switch( bs )
00137     {
00138         /* see SetBitsAllocated for explanation
00139         */
00140         case 0xffff: bs = 16; break;
00141         case 0x0fff: bs = 12; break;
00142         case 0x00ff: bs = 8; break;
00143     }
00144     if( bs <= BitsAllocated && bs )
00145     {
00146         BitsStored = bs;
00147         SetHighBit( (unsigned short) (bs - 1) );
00148     }
00149 }
00150
00152 unsigned short GetHighBit() const

```

```

00153 {
00154     gdcm_assert( HighBit < BitsStored );
00155     return HighBit;
00156 }
00157 void SetHighBit(unsigned short hb)
00158 {
00159     switch( hb )
00160     {
00161         /* broken implementations that use bitmask for BitsAllocated/Stored
00162          * nonetheless use (BitsStored-1) for HighBit. correct for this here.
00163          */
00164         case 0xfffe: hb = 15; break;
00165         case 0x0ffe: hb = 11; break;
00166         case 0x00fe: hb = 7; break;
00167     }
00168     if( BitsStored > 1 && hb == 0 )
00169         HighBit = BitsStored - 1;
00170     else if( hb < BitsStored )
00171         HighBit = hb;
00172 }
00173
00175 unsigned short GetPixelRepresentation() const
00176 {
00177     return (unsigned short)(PixelRepresentation ? 1 : 0);
00178 }
00179 void SetPixelRepresentation(unsigned short pr)
00180 {
00181     PixelRepresentation = (unsigned short)(pr ? 1 : 0);
00182 }
00183
00185 ScalarType GetScalarType() const;
00186
00189 void SetScalarType(ScalarType st);
00190 const char *GetScalarTypeAsString() const;
00191
00197 uint8_t GetPixelSize() const;
00198
00200 void Print(std::ostream &os) const;
00201
00203 int64_t GetMin() const;
00204
00206 int64_t GetMax() const;
00207
00209 bool IsValid() const;
00210
00211 bool operator==(ScalarType st) const
00212 {
00213     return GetScalarType() == st;
00214 }
00215 bool operator!=(ScalarType st) const
00216 {
00217     return GetScalarType() != st;
00218 }
00219 bool operator==(const PixelFormat &pf) const
00220 {
00221     return
00222         SamplesPerPixel == pf.SamplesPerPixel &&
00223         BitsAllocated == pf.BitsAllocated &&
00224         BitsStored == pf.BitsStored &&
00225         HighBit == pf.HighBit &&
00226         PixelRepresentation == pf.PixelRepresentation;
00227 }
00228 bool operator!=(const PixelFormat &pf) const
00229 {
00230     return
00231         SamplesPerPixel != pf.SamplesPerPixel ||
00232         BitsAllocated != pf.BitsAllocated ||
00233         BitsStored != pf.BitsStored ||
00234         HighBit != pf.HighBit ||
00235         PixelRepresentation != pf.PixelRepresentation;
00236 }
00237
00238 bool IsCompatible(const TransferSyntax & ts ) const;
00239 protected:
00241 bool Validate();
00242
00243 private:
00244     // D 0028|0002 [US] [Samples per Pixel] [1]
00245     unsigned short SamplesPerPixel;
00246     // D 0028|0100 [US] [Bits Allocated] [8]
00247     unsigned short BitsAllocated;

```

13.377 gdcmPixmap.h File Reference

[illegible]

```

graph TD
    gdcmPixmap_h[gdcmPixmap.h]
    gdcmFileDecompressLookupTable_h[gdcmFileDecompressLookupTable.h]
    gdcmIcnImageGenerator_h[gdcmIcnImageGenerator.h]
    gdcmPixmapReader_h[gdcmPixmapReader.h]
    gdcmImage_h[gdcmImage.h]
    gdcmPixmapWriter_h[gdcmPixmapWriter.h]
    gdcmImageReader_h[gdcmImageReader.h]
    gdcmSplitMosaicFilter_h[gdcmSplitMosaicFilter.h]
    gdcmImageWriter_h[gdcmImageWriter.h]
    gdcmImageRegionReader_h[gdcmImageRegionReader.h]

    gdcmPixmap_h --> gdcmFileDecompressLookupTable_h
    gdcmPixmap_h --> gdcmIcnImageGenerator_h
    gdcmPixmap_h --> gdcmPixmapReader_h
    gdcmPixmap_h --> gdcmImage_h
    gdcmPixmap_h --> gdcmPixmapWriter_h
    gdcmImageReader_h --> gdcmImage_h
    gdcmSplitMosaicFilter_h --> gdcmImage_h
    gdcmImageWriter_h --> gdcmImage_h
    gdcmImageRegionReader_h --> gdcmImageReader_h
    gdcmImageRegionReader_h --> gdcmImage_h
  
```

Classes

- class [gdcm::Pixmap](#)
Pixmap class.

Namespaces

- namespace [gdcm](#)

13.378 gdcmPixmap.h

[Go to the documentation of this file.](#)

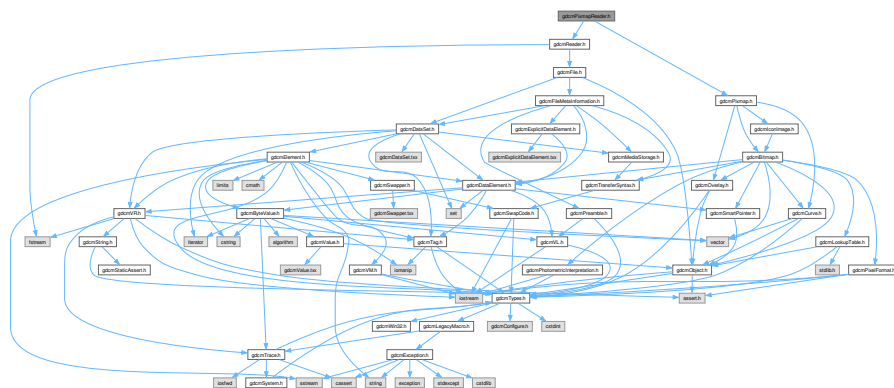
```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPIXMAP_H
00015 #define GDCMPIXMAP_H
00016
00017 #include "gdcmBitmap.h"
00018 #include "gdcmCurve.h"
00019 #include "gdcmIconImage.h"
00020 #include "gdcmOverlay.h"
00021
00022 namespace gdcm
00023 {
00024
00032 class GDCM_EXPORT Pixmap : public Bitmap
00033 {
00034 public:
00035   Pixmap();
00036   ~Pixmap() override;
00037   void Print(std::ostream &) const override;
00038
00040   bool AreOverlaysInPixelData() const override;
00042   bool UnusedBitsPresentInPixelData() const override;
00043
00045   Curve& GetCurve(size_t i = 0) {
00046     gdcm_assert( i < Curves.size() );
00047     return Curves[i];
00048   }
00049   const Curve& GetCurve(size_t i = 0) const {
00050     gdcm_assert( i < Curves.size() );
00051     return Curves[i];
00052   }
00053   size_t GetNumberOfCurves() const { return Curves.size(); }
00054   void SetNumberOfCurves(size_t n) { Curves.resize(n); }
00055
00057   Overlay& GetOverlay(size_t i = 0) {
00058     gdcm_assert( i < Overlays.size() );
00059     return Overlays[i];
00060   }
00061   const Overlay& GetOverlay(size_t i = 0) const {
00062     gdcm_assert( i < Overlays.size() );
00063     return Overlays[i];
00064   }
00065   size_t GetNumberOfOverlays() const { return Overlays.size(); }
00066   void SetNumberOfOverlays(size_t n) { Overlays.resize(n); }

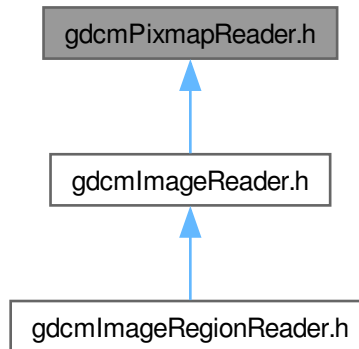
```

13.379 gdcmPixmapReader.h File Reference

Include dependency graph for gdcmPixmapReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::PixmapReader`
`PixmapReader`.

Namespaces

- namespace `gdcm`

13.380 gdcmPixmapReader.h

[Go to the documentation of this file.](#)

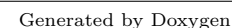
```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMPIXMAPREADER_H
00015  #define GDCMPIXMAPREADER_H
00016  00017  #include "gdcmReader.h"
00018  #include "gdcmPixmap.h"
00019  00020  namespace gdcm
00021  {

```

13.381 gdcmPixmapToPixmapFilter.h File Reference

Include dependency graph for `gdcmPixmapToPixmapFilter.h`:



Classes

- class [gdcm::PixmapToPixmapFilter](#)
[PixmapToPixmapFilter](#) class.

Namespaces

- namespace [gdcm](#)

13.382 gdcmPixmapToPixmapFilter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPIXMAPTOPIXMAPFILTER_H
00015  #define GDCMPIXMAPTOPIXMAPFILTER_H
00016
00017  #include "gdcmBitmapToBitmapFilter.h"
00018
00019  namespace gdcm
00020  {
00021
00022  class Pixmap;
00027  class GDCM_EXPORT PixmapToPixmapFilter : public BitmapToBitmapFilter
00028  {
00029  public:
00030  PixmapToPixmapFilter();
00031  ~PixmapToPixmapFilter() = default;
00032
00033  Pixmap &GetInput();
00034
00036  const Pixmap &GetOutput() const;
00037
00038  // SWIG/Java hack:
00039  const Pixmap &GetOutputAsPixmap() const;
00040  };
00041
00042  } // end namespace gdcm
00043
00044  #endif //GDCMPIXMAPTOPIXMAPFILTER_H

```

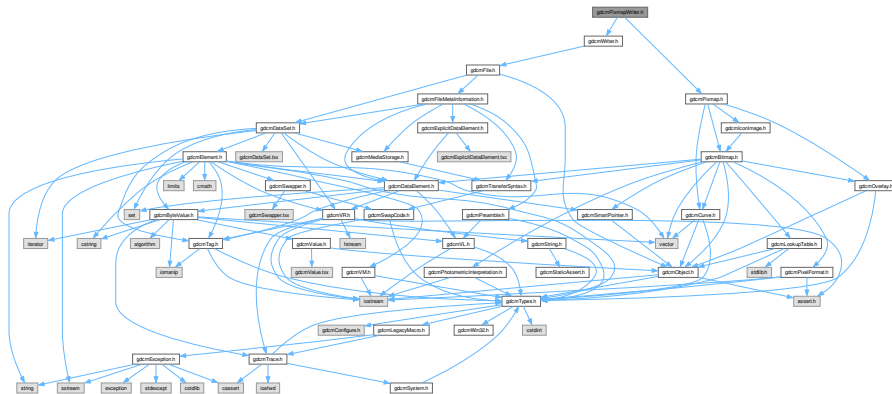
13.383 gdcmPixmapWriter.h File Reference

```

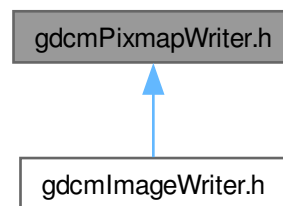
#include "gdcmWriter.h"
#include "gdcmPixmap.h"

```

Include dependency graph for `gdcmPixmapWriter.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::PixmapWriter`
`PixmapWriter`.

Namespaces

- namespace `gdcm`

13.384 gdcmPixmapWriter.h

[Go to the documentation of this file.](#)

00001

/*=====

```

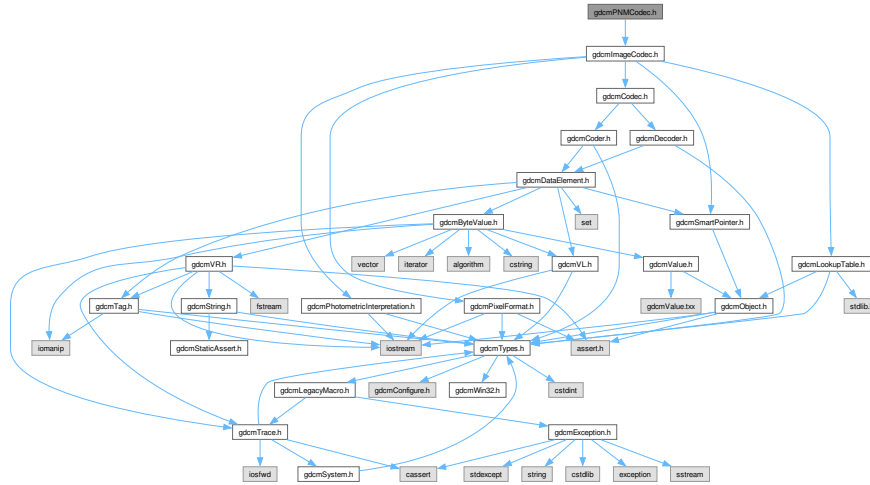
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMPIXMAPWRITER_H
00015 #define GDCMPIXMAPWRITER_H
00016
00017 #include "gdcmWriter.h"
00018 #include "gdcmPixmap.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class StreamImageWriter;
00024 class Pixmap;
00036 class GDCM_EXPORT PixmapWriter : public Writer
00037 {
00038 public:
00039 PixmapWriter();
00040 ~PixmapWriter() override;
00041
00042 const Pixmap& GetPixmap() const { return *PixelData; }
00043 Pixmap& GetPixmap() { return *PixelData; } // FIXME
00044 void SetPixmap(Pixmap const &img);
00045
00049 virtual const Pixmap& GetImage() const { return *PixelData; }
00050 virtual Pixmap& GetImage() { return *PixelData; } // FIXME
00051 virtual void SetImage(Pixmap const &img);
00052
00054 bool Write() override; // Execute()
00055
00056 protected:
00057 void DoIconImage(DataSet & ds, Pixmap const & image);
00058 bool PrepareWrite( MediaStorage const & refms );
00059
00060 SmartPointer<Pixmap> PixelData;
00061 };
00062
00067
00068 } // end namespace gdcm
00069
00070 #endif //GDCMPIXMAPWRITER_H

```

13.385 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPNMCodec.h:



Classes

- class [gdcm::PNMCodec](#)
Class to do PNM.

Namespaces

- namespace [gdcm](#)

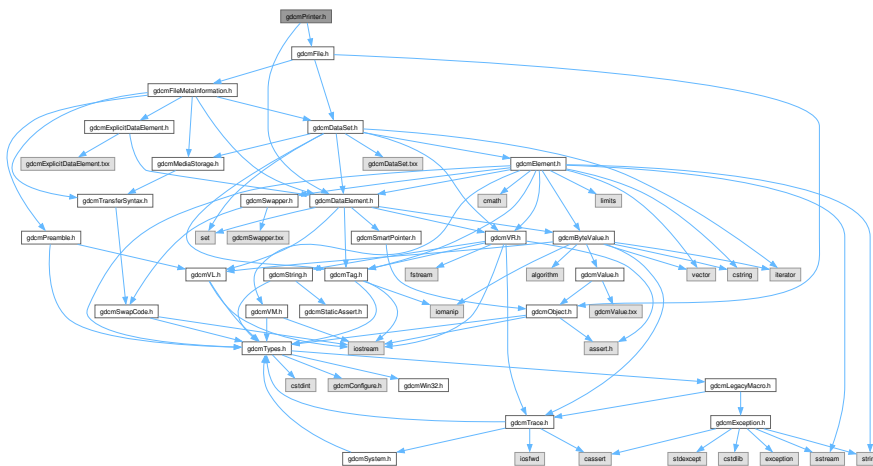
13.386 gdcmPNMCodec.h

[Go to the documentation of this file.](#)

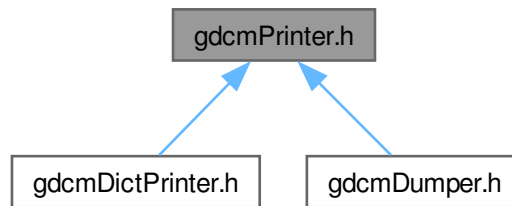
```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPNMCODEC_H
00015 #define GDCMPNMCODEC_H
00016
```

13.387 gdcmPrinter.h File Reference

Include dependency graph for gdcMPrinter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::Printer](#)
[Printer](#) class.

Namespaces

- namespace [gdcml](#)

13.388 gdcmlPrinter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPRINTER_H
00015  #define GDCMPRINTER_H
00016
00017  // TODO Class to implement printing
00018  // Since DICOM does printing ?
00019  // Also I would like to encapsulate the IsCharacterPrintable thing
00020  // (to avoid printing \0 and other weird characters)
00021  // \todo I still need to implement skipping of group (shadow)
00022  // need to implement longer field to read
00023
00024  /*
00025  * Output:
00026  * For ASCII:
00027  * Typically will look like:
00028  * [ORIGINAL\PRIMARY\OTHER]
  
```



```

00029 * If a non printable character is found: RED and INVERSE is used:
00030 * [          .]
00031 *
00032 * when the VR is not found (file or dict), we check if we can print the output:
00033 * on success ASCII mode is used, on failure the output is printed a series of bytes
00034 *
00035 * Special case when the data element is empty:
00036 * INVERSE « (no value)
00037 *
00038 * retired public element are printed in red and underline
00039 * unknown private element are printed in RED followed by 'UNKNOWN'
00040 *
00041 * Correct VR is printed in green just after the found VR
00042 *
00043 * length of data element is printed in bytes, followed by the VM, a green VM is appended
00044 * if this is not compatible
00045 */
00046 #include "gdcmFile.h"
00047 #include "gdcmDataElement.h"
00048
00049 namespace gdcm
00050 {
00051
00052 class DataSet;
00053 class DictEntry;
00054 class Dicts;
00055 // It's a sink there is no output
00056 class GDCM_EXPORT Printer
00057 {
00058 public:
00059 Printer();
00060 ~Printer() = default;
00061
00062 void SetFile(File const &f) { F = &f; }
00063
00064 void SetColor(bool c);
00065
00066 typedef enum {
00067     VERBOSE_STYLE = 0, // GDCM Legacy VERBOSE one
00068     CONDENSED_STYLE, //
00069     // Ok I am missing voc here ...better naming would be nice
00070     XML, //
00071     CXX
00072 } PrintStyles;
00073
00074 void SetStyle(PrintStyles ps) {
00075     PrintStyle = ps;
00076 }
00077
00078 PrintStyles GetPrintStyle() const {
00079     return PrintStyle;
00080 }
00081
00082 void Print(std::ostream& os);
00083
00084 void PrintDataSet(const DataSet &ds, std::ostream& os, const std::string &s = "");
00085
00086 protected:
00087 VR PrintDataElement(std::ostream& os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream
&out, std::string const &indent );
00088 void PrintSQ(const SequenceOfItems *sqi, std::ostream &os, std::string const &indent);
00089
00090 PrintStyles PrintStyle;
00091 const File *F;
00092 VL MaxPrintLength;
00093 };
00094
00095 } // end namespace gdcm
00096
00097 #endif //GDCMPRINTER_H

```



```

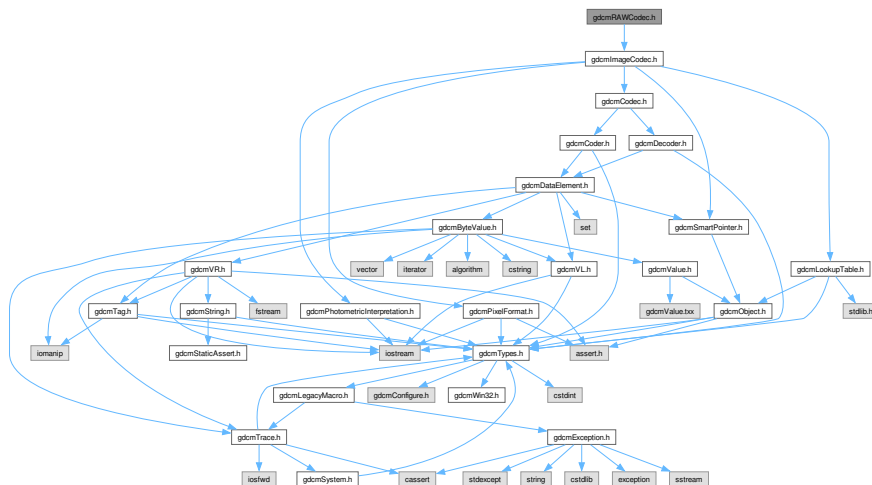
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00034 class PVRGCodec : public ImageCodec
00035 {
00036 public:
00037   PVRGCodec();
00038   ~PVRGCodec() override;
00039   bool CanDecode(TransferSyntax const &ts) const override;
00040   bool CanCode(TransferSyntax const &ts) const override;
00041
00042   bool Decode(DataElement const &is, DataElement &os) override;
00043   bool Code(DataElement const &in, DataElement &out) override;
00044   void SetLossyFlag( bool l );
00045
00046   ImageCodec * Clone() const override;
00047 private:
00048 };
00049
00050 } // end namespace gdcm
00051
00052 #endif //GDCMPVRGCODEC_H

```

13.391 gdcmRAWCodec.h File Reference

#include "gdcmImageCodec.h"

Include dependency graph for gdcmRAWCodec.h:



Classes

- class [gdcm::RAWCodec](#)
RAWCodec class.

Namespaces

- namespace [gdcm](#)

13.392 gdcmRAWCodec.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMRAWCODEC_H
00015 #define GDCMRAWCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class RAWInternals;
00026 class GDCM_EXPORT RAWCodec : public ImageCodec
00027 {
00028 public:
00029 RAWCodec();
00030 ~RAWCodec() override;
00031 bool CanCode(TransferSyntax const &ts) const override;
00032 bool CanDecode(TransferSyntax const &ts) const override;
00033 bool Decode(DataElement const &is, DataElement &os) override;
00034 bool Code(DataElement const &in, DataElement &out) override;
00035
00036 bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00037 ImageCodec * Clone() const override;
00038
00041 bool DecodeBytes(const char* inBytes, size_t inBufferLength,
00042 char* outBytes, size_t inOutBufferLength);
00043
00044 protected:
00045 bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00046
00047 private:
00048 RAWInternals *Internals;
00049 };
00050
00051 } // end namespace gdcm
00052
00053 #endif // GDCMRAWCODEC_H

```

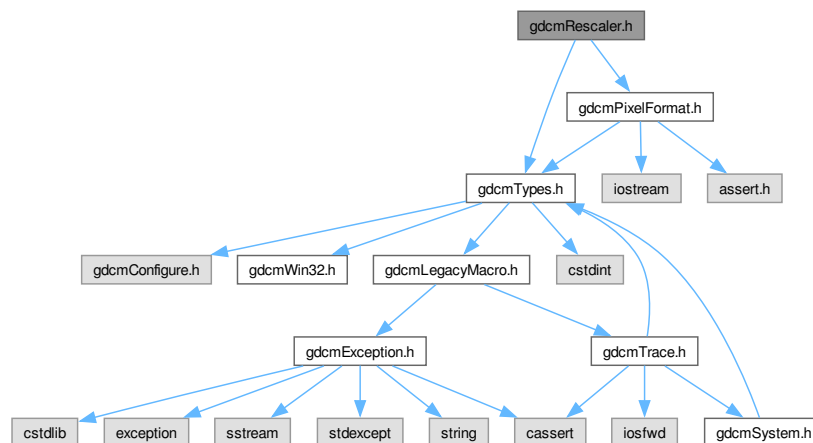
13.393 gdcmRescaler.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"

```

Include dependency graph for gdcmRescaler.h:



Classes

- class [gdcm::Rescaler](#)
Rescale class.

Namespaces

- namespace [gdcm](#)

13.394 gdcmRescaler.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMRESCALER_H
00015  #define GDCMRESCALER_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmPixelFormat.h"
00019  00020  namespace gdcm
00021  {

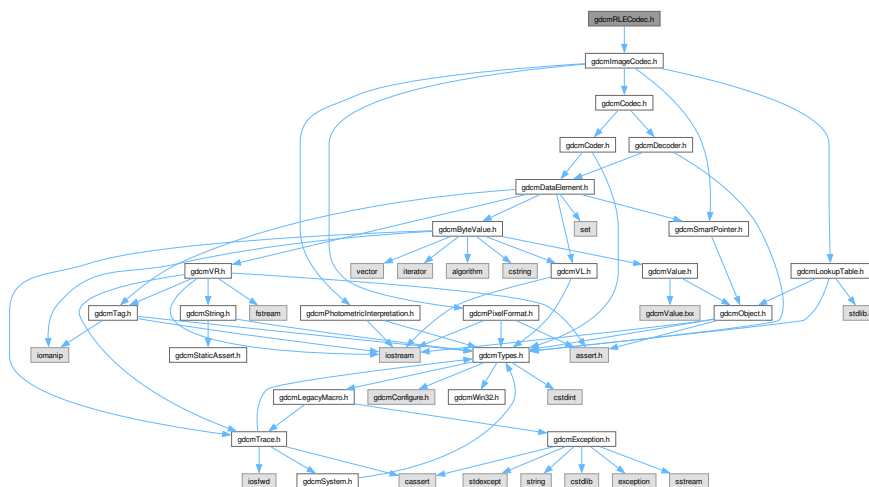
```

```

00022
00068 class GDCM_EXPORT Rescaler
00069 {
00070 public:
00071     Rescaler():Intercept(0),Slope(1),PF(PixelFormat::UNKNOWN),TargetScalarType(PixelFormat::UNKNOWN),
        ScalarRangeMin(0), ScalarRangeMax(0), UseTargetPixelType(false) {}
00072     ~Rescaler() = default;
00073
00075     bool Rescale(char *out, const char *in, size_t n);
00076
00078     bool InverseRescale(char *out, const char *in, size_t n);
00079
00081     void SetIntercept(double i) { Intercept = i; }
00082     double GetIntercept() const { return Intercept; }
00083
00085     void SetSlope(double s) { Slope = s; }
00086     double GetSlope() const { return Slope; }
00087
00092     void SetTargetPixelType( PixelFormat const & targetst );
00093
00095     void SetUseTargetPixelType(bool b);
00096
00098     void SetPixelFormat(PixelFormat const & pf) { PF = pf; }
00099
00102     PixelFormat::ScalarType ComputeInterceptSlopePixelType();
00103
00106     void SetMinMaxForPixelType(double min, double max);
00107
00110     PixelFormat ComputePixelTypeFromMinMax();
00111
00112 protected:
00113     template <typename TIn>
00114         void RescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
00115     template <typename TIn>
00116         void InverseRescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
00117
00118 private:
00119     double Intercept; // 0028,1052
00120     double Slope; // 0028,1053
00121     PixelFormat PF;
00122     PixelFormat::ScalarType TargetScalarType;
00123     double ScalarRangeMin;
00124     double ScalarRangeMax;
00125     bool UseTargetPixelType;
00126 };
00127
00128 } // end namespace gdcm
00129
00130 #endif //GDCMRESCALER_H

```

```
#include "gdcmImageCodec.h"
Include dependency graph for gdcmRLECodec.h:
```



- class `gdc::RLECodec`
Class to do RLE.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```
00001  /*=====
00002  /
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMRLECODEC_H
00015 #define GDCMRLECODEC_H
00016
```

```

00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class Fragment;
00023 class RLEInternals;
00036 class GDCM_EXPORT RLECodec : public ImageCodec
00037 {
00038 friend class ImageRegionReader;
00039 public:
00040   RLECodec();
00041   ~RLECodec() override;
00042   bool CanCode(TransferSyntax const &ts) const override;
00043   bool CanDecode(TransferSyntax const &ts) const override;
00044   bool Decode(DataElement const &is, DataElement &os) override;
00045   unsigned long GetBufferLength() const { return BufferLength; }
00046   void SetBufferLength(unsigned long l) { BufferLength = l; }
00047
00048   bool Code(DataElement const &in, DataElement &out) override;
00049   bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
00050   ImageCodec * Clone() const override;
00051
00052 protected:
00053   bool DecodeExtent(
00054     char *buffer,
00055     unsigned int XMin, unsigned int XMax,
00056     unsigned int YMin, unsigned int YMax,
00057     unsigned int ZMin, unsigned int ZMax,
00058     std::istream & is
00059   );
00060
00061   bool DecodeByStreams(std::istream &is, std::ostream &os) override;
00062 public:
00063
00064   void SetLength(unsigned long l)
00065   {
00066     Length = l;
00067   }
00068
00069 protected:
00070   bool StartEncode( std::ostream & ) override;
00071   bool IsRowEncoder() override;
00072   bool IsFrameEncoder() override;
00073   bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
00074   bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
00075   bool StopEncode( std::ostream & ) override;
00076
00077 private:
00078   bool DecodeByStreamsCommon(std::istream &is, std::ostream &os);
00079   RLEInternals *Internals;
00080   unsigned long Length;
00081   unsigned long BufferLength;
00082   size_t DecodeFragment(Fragment const & frag, char *buffer, size_t llen);
00083 };
00084
00085 } // end namespace gdcm
00086
00087 #endif //GDCMRLECODEC_H

```

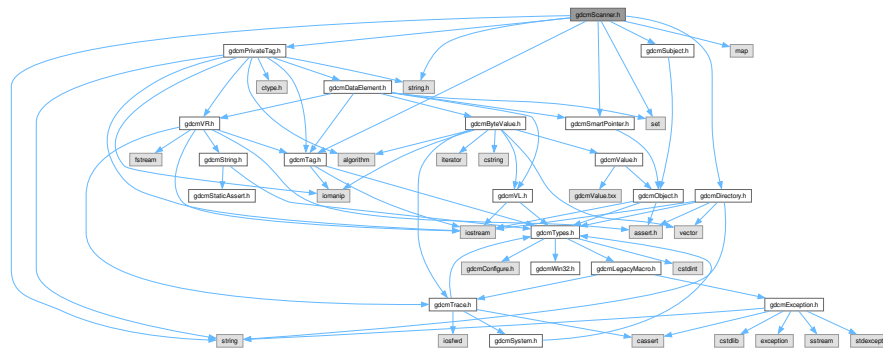
13.397 gdcmScanner.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>

```


Include dependency graph for gdcmsScanner.h:



- struct `gdcm::Scanner::ltstr`
- class `gdcm::Scanner`
`Scanner.`

- namespace `gdcm`

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Scanner &s)`

Go to the documentation of this file.

```

00001  /*=====
00002  /
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSCANNER_H
00015 #define GDCMSCANNER_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmSubject.h"

```

```

00019 #include "gdcmTag.h"
00020 #include "gdcmPrivateTag.h"
00021 #include "gdcmSmartPointer.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcm
00030 {
00031 class StringFilter;
00032
00033 class GDCM_EXPORT Scanner : public Subject
00034 {
00035     friend std::ostream& operator<<(std::ostream &_os, const Scanner &s);
00036 public:
00037     Scanner():Values(),Filenames(),Mappings(),Progress(0.0) {}
00038     ~Scanner() override;
00039
00040     typedef std::map<Tag, const char*> TagToValue;
00041     //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
00042     //typedef TagToStringMap TagToValue;
00043     typedef TagToValue::value_type TagToValueValueType;
00044
00045     void AddTag( Tag const & t );
00046     void ClearTags();
00047
00048     // Work in progress do not use:
00049     void AddPrivateTag( PrivateTag const & t );
00050
00051     void AddSkipTag( Tag const & t );
00052     void ClearSkipTags();
00053
00054     bool Scan( Directory::FilenamesType const & filenames );
00055
00056     Directory::FilenamesType const &GetFilenames() const { return Filenames; }
00057
00058     void Print( std::ostream & os ) const override;
00059
00060     void PrintTable( std::ostream & os ) const;
00061
00062     bool IsKey( const char * filename ) const;
00063
00064     Directory::FilenamesType GetKeys() const;
00065
00066     // struct to store all the values found:
00067     typedef std::set< std::string > ValuesType;
00068
00069     ValuesType const & GetValues() const { return Values; }
00070
00071     ValuesType GetValues(Tag const &t) const;
00072
00073     Directory::FilenamesType GetOrderedValues(Tag const &t) const;
00074
00075     /* Itstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00076     struct Itstr
00077     {
00078         bool operator()(const char* s1, const char* s2) const
00079         {
00080             gdcm_assert( s1 && s2 );
00081             return strcmp(s1, s2) < 0;
00082         }
00083     };
00084
00085     typedef std::map<const char *,TagToValue, Itstr> MappingType;
00086     typedef MappingType::const_iterator ConstIterator;
00087     ConstIterator Begin() const { return Mappings.begin(); }
00088     ConstIterator End() const { return Mappings.end(); }
00089
00090     MappingType const & GetMappings() const { return Mappings; }
00091
00092     TagToValue const & GetMapping(const char *filename) const;
00093
00094     const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00095
00096     Directory::FilenamesType GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const;
00097
00098     // by a call to GetMapping()
00099     TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00100
00101

```

```

00154 const char* GetValue(const char *filename, Tag const &t) const;
00155
00157 static SmartPointer<Scanner> New() { return new Scanner; }
00158
00159 protected:
00160 void ProcessPublicTag(StringFilter &sf, const char *filename);
00161 private:
00162 // struct to store all uniq tags in ascending order:
00163 typedef std::set< Tag > TagsType;
00164 typedef std::set< PrivateTag > PrivateTagsType;
00165 std::set< Tag > Tags;
00166 std::set< PrivateTag > PrivateTags;
00167 std::set< Tag > SkipTags;
00168 ValuesType Values;
00169 Directory::FileNamesType Filenames;
00170
00171 // Main struct that will hold all mapping:
00172 MappingType Mappings;
00173
00174 double Progress;
00175 };
00176 //-----
00177 inline std::ostream& operator<<(std::ostream &os, const Scanner &s)
00178 {
00179 s.Print( os );
00180 return os;
00181 }
00182
00183 #if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
00184 /*
00185 * HACK: I need this temp class to be able to manipulate a std::map from python,
00186 * swig does not support wrapping of simple class like std::map...
00187 */
00188 class SWIGTagToValue
00189 {
00190 public:
00191 SWIGTagToValue(Scanner::TagToValue const &t2v):Internal(t2v),it(t2v.begin()) {}
00192 const Scanner::TagToValueValueType& GetCurrent() const { return *it; }
00193 const Tag& GetCurrentTag() const { return it->first; }
00194 const char *GetCurrentValue() const { return it->second; }
00195 void Start() { it = Internal.begin(); }
00196 bool IsAtEnd() const { return it == Internal.end(); }
00197 void Next() { ++it; }
00198 private:
00199 const Scanner::TagToValue& Internal;
00200 Scanner::TagToValue::const_iterator it;
00201 };
00202 #endif /* SWIG */
00203
00208
00209 } // end namespace gdcmscanner
00210
00211 #endif //GDCMSCANNER_H

```

13.399 gdcmscanner2.h File Reference

```

#include "gdcmscanner2.h"
#include "gdcmscanner2.h"
#include "gdcmscanner2.h"
#include "gdcmscanner2.h"
#include "gdcmscanner2.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```



```

00021 #include "gdcmsmartpointer.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcmscanner
00030 {
00031     class StringFilter;
00032
00033     class GDCM_EXPORT Scanner2 : public Subject
00034     {
00035     public:
00036         friend std::ostream& operator<<(std::ostream &_os, const Scanner2 &s);
00037     public:
00038         Scanner2():Values(),FileNames(),PublicMappings(),PrivateMappings(),Progress(0.0) {}
00039         ~Scanner2() override;
00040
00041         typedef std::map<Tag, const char*> PublicTagToValue;
00042         typedef PublicTagToValue::value_type PublicTagToValueValueType;
00043
00044         typedef std::map<PrivateTag, const char*> PrivateTagToValue;
00045         typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
00046
00047         bool AddPublicTag( Tag const & t );
00048         void ClearPublicTags();
00049
00050         // Work in progress do not use:
00051         bool AddPrivateTag( PrivateTag const & pt );
00052         void ClearPrivateTags();
00053
00054         bool AddSkipTag( Tag const & t );
00055         void ClearSkipTags();
00056
00057         bool Scan( Directory::FileNamesType const & filenames );
00058
00059         Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00060
00061         void Print( std::ostream & os ) const override;
00062
00063         void PrintTable( std::ostream & os, bool header = false ) const;
00064
00065         bool IsKey( const char * filename ) const;
00066
00067         Directory::FileNamesType GetKeys() const;
00068
00069         // struct to store all the values found:
00070         typedef std::set< std::string > ValuesType;
00071
00072         ValuesType const & GetValues() const { return Values; }
00073
00074         ValuesType GetPublicValues(Tag const &t) const;
00075
00076         ValuesType GetPrivateValues(PrivateTag const &pt) const;
00077
00078         Directory::FileNamesType GetPublicOrderedValues(Tag const &t) const;
00079
00080         Directory::FileNamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
00081
00082         /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00083         struct ltstr
00084         {
00085         public:
00086             bool operator()(const char* s1, const char* s2) const
00087             {
00088                 gdcmscanner::assert( s1 && s2 );
00089                 return strcmp(s1, s2) < 0;
00090             }
00091         };
00092
00093         typedef std::map<const char *,PublicTagToValue, ltstr> PublicMappingType;
00094         typedef PublicMappingType::const_iterator PublicConstIterator;
00095         PublicConstIterator Begin() const { return PublicMappings.begin(); }
00096         PublicConstIterator End() const { return PublicMappings.end(); }
00097
00098         typedef std::map<const char *,PrivateTagToValue, ltstr> PrivateMappingType;
00099         typedef PrivateMappingType::const_iterator PrivateConstIterator;
00100         PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
00101         PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }
00102
00103         PublicMappingType const & GetPublicMappings() const { return PublicMappings; }
00104         PrivateMappingType const & GetPrivateMappings() const { return PrivateMappings; }

```

```

00148
00150 PublicTagToValue const & GetPublicMapping(const char *filename) const;
00151 PrivateTagToValue const & GetPrivateMapping(const char *filename) const;
00152
00155 const char *GetFilenameFromPublicTagToValue(Tag const &t, const char *valueref) const;
00156 const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt, const char *valueref) const;
00157
00160 Directory::FileNamesType GetAllFileNamesFromPublicTagToValue(Tag const &t, const char *valueref) const;
00161 Directory::FileNamesType GetAllFileNamesFromPrivateTagToValue(PrivateTag const &pt, const char *valueref) const;
00162
00164 // by a call to GetMapping()
00165 PublicTagToValue const & GetMappingFromPublicTagToValue(Tag const &t, const char *value) const;
00166 PrivateTagToValue const & GetMappingFromPrivateTagToValue(PrivateTag const &pt, const char *value) const;
00167
00173 const char* GetPublicValue(const char *filename, Tag const &t) const;
00174 const char* GetPrivateValue(const char *filename, PrivateTag const &t) const;
00175
00177 static SmartPointer<Scanner2> New() { return new Scanner2; }
00178
00179 protected:
00180 void ProcessPublicTag(StringFilter &sf, const char *filename);
00181 void ProcessPrivateTag(StringFilter &sf, const char *filename);
00182 private:
00183 // struct to store all uniq tags in ascending order:
00184 typedef std::set< Tag > PublicTagsType;
00185 typedef std::set< PrivateTag > PrivateTagsType;
00186 std::set< Tag > PublicTags; // Public and Private Creator
00187 std::set< PrivateTag > PrivateTags; // Only Private (no Private Creator)
00188 std::set< Tag > SkipTags;
00189 ValuesType Values;
00190 Directory::FileNamesType FileNames;
00191
00192 // Main struct that will hold all public mapping:
00193 PublicMappingType PublicMappings;
00194 // Main struct that will hold all private mapping:
00195 PrivateMappingType PrivateMappings;
00196
00197 double Progress;
00198 };
00199 //-----
00200 inline std::ostream& operator<<(std::ostream &os, const Scanner2 &s)
00201 {
00202     s.Print( os );
00203     return os;
00204 }
00205
00206 } // end namespace gdcmm
00207
00208 #endif //GDCMSCANNER2_H

```

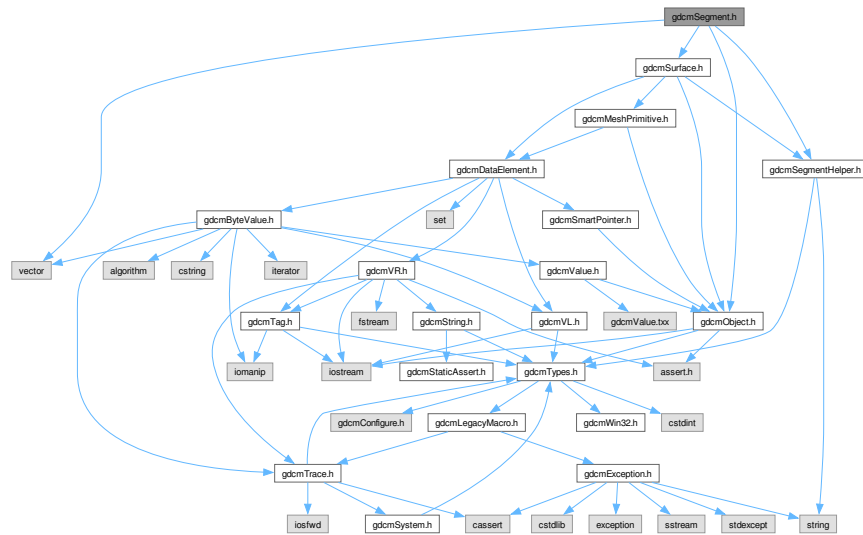
13.401 gdcmmSegment.h File Reference

```

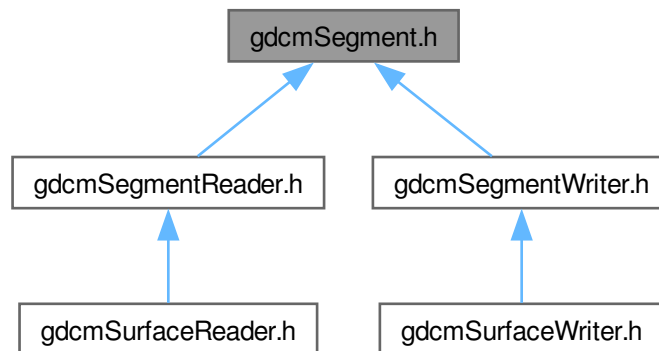
#include <vector>
#include <gdcmmObject.h>
#include <gdcmmSurface.h>
#include "gdcmmSegmentHelper.h"

```

Include dependency graph for gdcmSegment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::Segment](#)
This class defines a segment.

Namespaces

- namespace [gdcm](#)

13.402 gdcSegment.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSEGMENT_H
00015 #define GDCMSEGMENT_H
00016
00017 #include <vector>
00018
00019 #include <gdcObject.h>
00020 #include <gdcSurface.h>
00021 #include "gdcSegmentHelper.h"
00022
00023 namespace gdc
00024 {
00025
00033 class GDCM_EXPORT Segment : public Object
00034 {
00035 public:
00036
00037     typedef std::vector< SmartPointer< Surface > > SurfaceVector;
00038     typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector;
00039
00040     typedef enum {
00041         AUTOMATIC = 0,
00042         SEMIAUTOMATIC,
00043         MANUAL,
00044         ALGOTYPE_END
00045     } ALGOTYPE;
00046
00047     static const char * GetALGOTYPEString(ALGOTYPE type);
00048     static ALGOTYPE GetALGOTYPE(const char * type);
00049
00050
00051     Segment();
00052
00053     ~Segment() override;
00054
00055     /**      Segment getters/setters      **/
00056     unsigned short GetSegmentNumber() const;
00057     void SetSegmentNumber(const unsigned short num);
00058
00059     const char * GetSegmentLabel() const;
00060     void SetSegmentLabel(const char * label);
00061
00062     const char * GetSegmentDescription() const;
00063     void SetSegmentDescription(const char * description);
00064
00065     SegmentHelper::BasicCodedEntry const & GetAnatomicRegion() const;
00066     SegmentHelper::BasicCodedEntry & GetAnatomicRegion();
00067     void SetAnatomicRegion(SegmentHelper::BasicCodedEntry const & BSE);
00068
00069     BasicCodedEntryVector const & GetAnatomicRegionModifiers() const;
00070     BasicCodedEntryVector & GetAnatomicRegionModifiers();
00071     void SetAnatomicRegionModifiers(BasicCodedEntryVector const & BSEV);
00072
00073     SegmentHelper::BasicCodedEntry const & GetPropertyCategory() const;
00074     SegmentHelper::BasicCodedEntry & GetPropertyCategory();
00075     void SetPropertyCategory(SegmentHelper::BasicCodedEntry const & BSE);
00076
00077     SegmentHelper::BasicCodedEntry const & GetPropertyType() const;
00078     SegmentHelper::BasicCodedEntry & GetPropertyType();
00079     void SetPropertyType(SegmentHelper::BasicCodedEntry const & BSE);
00080

```



```

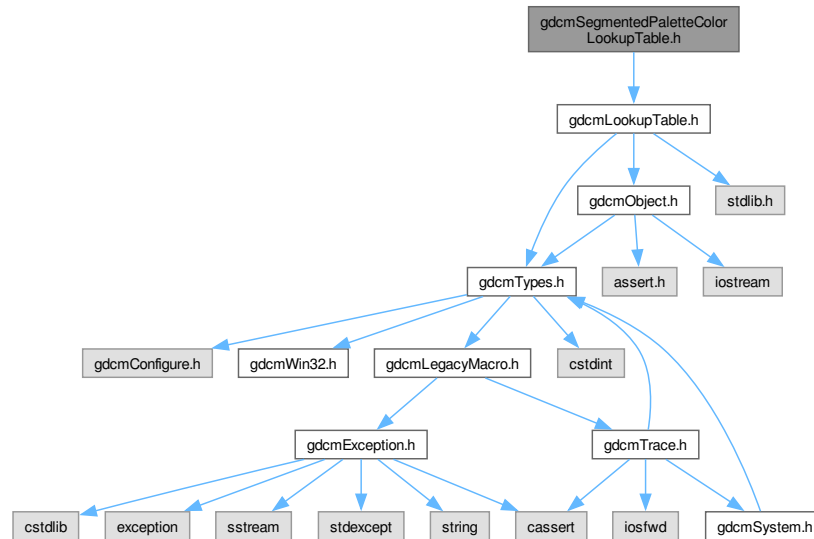
00081 BasicCodedEntryVector const & GetPropertyTypeModifiers() const;
00082 BasicCodedEntryVector & GetPropertyTypeModifiers();
00083 void SetPropertyTypeModifiers(BasicCodedEntryVector const & BSEV);
00084
00085 ALGOType GetSegmentAlgorithmType() const;
00086 void SetSegmentAlgorithmType(ALGOType type);
00087 void SetSegmentAlgorithmType(const char * typeStr);
00088
00089 const char * GetSegmentAlgorithmName() const;
00090 void SetSegmentAlgorithmName(const char * name);
00091
00092 /**      Surface getters/setters      **/
00093 unsigned long GetSurfaceCount();
00094 void SetSurfaceCount(const unsigned long nb);
00095
00096 SurfaceVector const & GetSurfaces() const;
00097 SurfaceVector & GetSurfaces();
00098
00099 SmartPointer< Surface > GetSurface(const unsigned int idx = 0) const;
00100
00101 void AddSurface(SmartPointer< Surface > surface);
00102
00103 protected :
00104 /**      Segment members      **/
00105 //0062 0004 US 1 Segment Number
00106 unsigned short SegmentNumber;
00107 //0062 0005 LO 1 Segment Label
00108 std::string SegmentLabel;
00109 //0062 0006 ST 1 Segment Description
00110 std::string SegmentDescription;
00111
00112 // General Anatomic Region
00113 SegmentHelper::BasicCodedEntry AnatomicRegion;
00114 // General Anatomic Region Modifier
00115 BasicCodedEntryVector AnatomicRegionModifiers;
00116 // Property Category Code
00117 SegmentHelper::BasicCodedEntry PropertyCategory;
00118 // Property Type Code
00119 SegmentHelper::BasicCodedEntry PropertyType;
00120 // Property Type Modifier Code
00121 BasicCodedEntryVector PropertyTypeModifiers;
00122
00123 //0062 0008 CS 1 Segment Algorithm Type
00124 ALGOType SegmentAlgorithmType;
00125 //0062 0009 LO 1 Segment Algorithm Name
00126 std::string SegmentAlgorithmName;
00127
00128 /**      Surface members      **/
00129 //0066 002a UL 1 Surface Count
00130 unsigned long SurfaceCount;
00131
00132 SurfaceVector Surfaces;
00133
00134 private :
00135 void ComputeSurfaceCount();
00136 };
00137
00138 }
00139
00140 #endif // GDCMSEGMENT_H

```

13.403 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)
SegmentedPaletteColorLookupTable class.

Namespaces

- namespace [gdcm](#)

13.404 gdcmSegmentedPaletteColorLookupTable.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012

```

```

00013  ===== */
00014
00015 #ifndef GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
00016 #define GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
00017
00018 #include "gdcmLookupTable.h"
00019
00020 namespace gdcm
00021 {
00022
00026 class GDCM_EXPORT SegmentedPaletteColorLookupTable : public LookupTable
00027 {
00028 public:
00029     SegmentedPaletteColorLookupTable();
00030     ~SegmentedPaletteColorLookupTable() override;
00031     void Print(std::ostream &) const override {}
00032
00034     void SetLUT(LookupTableType type, const unsigned char *array,
00035                unsigned int length) override;
00036
00037 };
00038
00039 } // end namespace gdcm
00040
00041 #endif //GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H

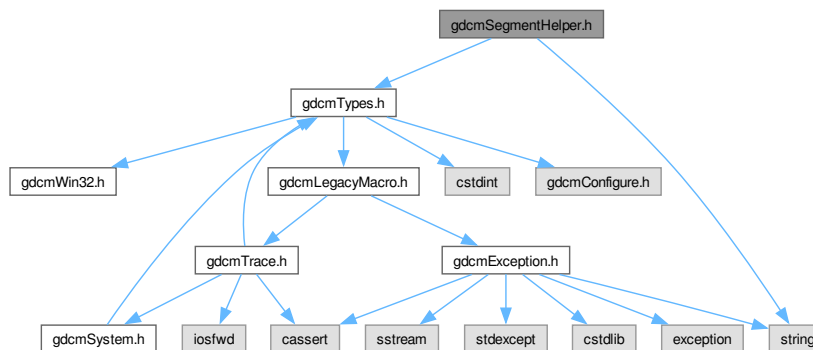
```

13.405 gdcmSegmentHelper.h File Reference

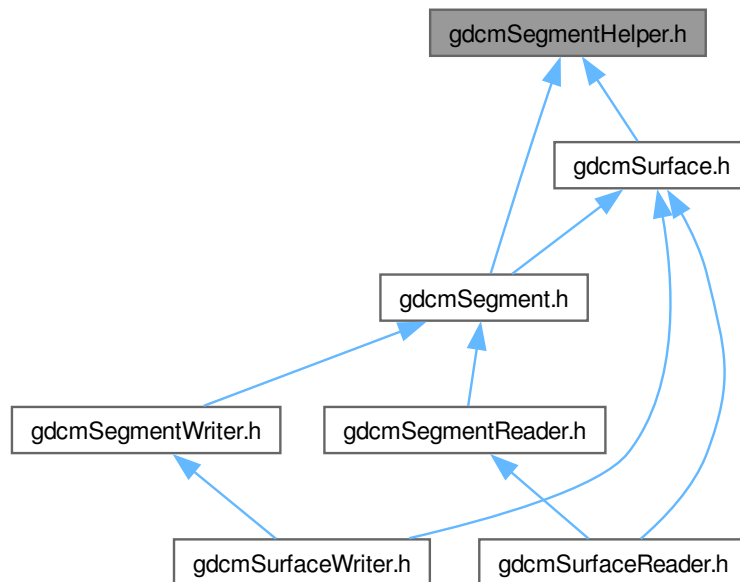
```
#include "gdcmTypes.h"
```

```
#include <string>
```

Include dependency graph for gdcmSegmentHelper.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [gdcm::SegmentHelper::BasicCodedEntry](#)
This structure defines a basic coded entry with all of its attributes.

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::SegmentHelper](#)

13.406 gdcmSegmentHelper.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.

```

```

00012
00013 =====*/
00014 #ifndef GDCMSEGMENTHELPER_H
00015 #define GDCMSEGMENTHELPER_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <string>
00020
00021 namespace gdcm
00022 {
00023
00024     namespace SegmentHelper
00025     {
00026
00032         struct GDCM_EXPORT BasicCodedEntry
00033         {
00037             BasicCodedEntry():
00038                 CV(""),
00039                 CSD(""),
00040                 CSV(""),
00041                 CM("")
00042             {}
00043
00047             BasicCodedEntry(const char * a_CV,
00048                             const char * a_CSD,
00049                             const char * a_CM):
00050                 CV(a_CV),
00051                 CSD(a_CSD),
00052                 CSV(""),
00053                 CM(a_CM)
00054             {}
00055
00059             BasicCodedEntry(const char * a_CV,
00060                             const char * a_CSD,
00061                             const char * a_CSV,
00062                             const char * a_CM):
00063                 CV(a_CV),
00064                 CSD(a_CSD),
00065                 CSV(a_CSV),
00066                 CM(a_CM)
00067             {}
00068
00074             bool IsEmpty(const bool checkOptionalAttributes = false) const;
00075
00076             /**    Members    */
00077             // 0008 0100 1   Code Value
00079             std::string CV;
00080             // 0008 0102 1   Coding Scheme Designator
00081             std::string CSD;
00082             // 0008 0103 1C   Coding Scheme Version
00083             std::string CSV;
00084             // 0008 0104 1   Code Meaning
00085             std::string CM;
00086         };
00087
00088     } // end of SegmentHelper namespace
00089
00090 } // end of gdcm namespace
00091
00092 #endif // GDCMSEGMENTHELPER_H

```

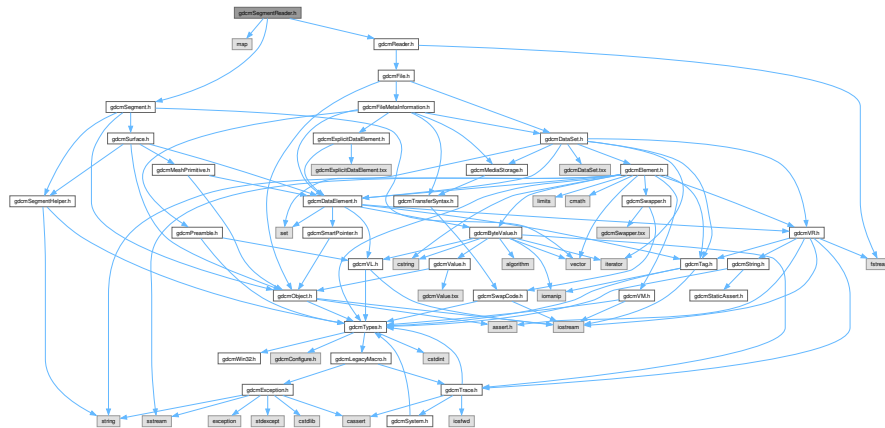
13.407 gdcmSegmentReader.h File Reference

```

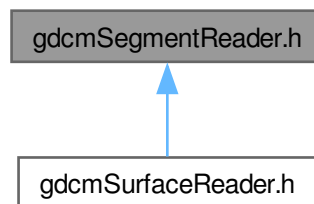
#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>

```

Include dependency graph for gdcmSegmentReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::SegmentReader`
This class defines a segment reader.

Namespaces

- namespace `gdcm`

13.408 gdcmSegmentReader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSEGMENTREADER_H
00015 #define GDCMSEGMENTREADER_H
00016
00017 #include <map>
00018
00019 #include <gdcmReader.h>
00020 #include <gdcmSegment.h>
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT SegmentReader : public Reader
00026 {
00027 public:
00028     typedef std::vector<SmartPointer<Segment>> SegmentVector;
00029
00030     SegmentReader();
00031
00032     ~SegmentReader() override;
00033
00034     bool Read() override; // Set to protected ?
00035
00036     /** Segment getters/setters */
00037     SegmentVector GetSegments() const;
00038     SegmentVector GetSegments();
00039
00040     // unsigned int GetNumberOfSegments();
00041
00042 protected:
00043
00044     typedef std::map<unsigned long, SmartPointer<Segment>> SegmentMap;
00045
00046     bool ReadSegments();
00047
00048     bool ReadSegment(const Item & segmentItem, const unsigned int idx);
00049
00050     SegmentMap Segments; // The key value is item number (in segment sequence)
00051                          // or the surface number (for a surface segmentation).
00052 };
00053
00054 }
00055
00056 #endif // GDCMSEGMENTREADER_H

```

13.409 gdcmSegmentWriter.h File Reference

```

#include <gdcmWriter.h>
#include <gdcmSegment.h>

```

[illegible]

```
graph BT; gdcmsurfacewriter[hgdcmsurfaceWriter.h] --> gdcmsegmentwriter[hgdcmsegmentWriter.h];
```

- class `gdcm::SegmentWriter`
This class defines a segment writer.

- namespace `gdcm`

13.410 gdcmSegmentWriter.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSEGMENTWRITER_H
00015 #define GDCMSEGMENTWRITER_H
00016
00017 #include <gdcmWriter.h>
00018 #include <gdcmSegment.h>
00019
00020 namespace gdcm
00021 {
00022
00023 class GDCM_EXPORT SegmentWriter : public Writer
00024 {
00025 public:
00026     typedef std::vector<SmartPointer<Segment>> SegmentVector;
00027
00028     SegmentWriter();
00029     ~SegmentWriter() override;
00030
00031     bool Write() override; // Set to protected ?
00032
00033     /** Segment getters/setters **/
00034     unsigned int GetNumberOfSegments() const;
00035     void SetNumberOfSegments(const unsigned int size);
00036
00037     const SegmentVector & GetSegments() const;
00038     SegmentVector & GetSegments();
00039     SmartPointer<Segment> GetSegment(const unsigned int idx = 0) const;
00040
00041     void AddSegment(SmartPointer<Segment> segment);
00042
00043     void SetSegments(SegmentVector & segments);
00044
00045 protected:
00046     bool PrepareWrite();
00047
00048     SegmentVector Segments;
00049 };
00050
00051 }
00052
00053 #endif // GDCMSEGMENTWRITER_H

```

13.411 gdcmSerieHelper.h File Reference

```

#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>

```


13.412 gdcmserieHelper.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSERIEHELPER_H
00015 #define GDCMSERIEHELPER_H
00016
00017 #include "gdcmsTag.h"
00018 #include "gdcmsSmartPointer.h"
00019 #include "gdcmsFile.h"
00020 #include <vector>
00021 #include <string>
00022 #include <map>
00023
00024 namespace gdcms
00025 {
00026
00027 enum CompOperators {
00028     GDCM_EQUAL = 0,
00029     GDCM_DIFFERENT,
00030     GDCM_GREATER,
00031     GDCM_GREATEROREQUAL,
00032     GDCM_LESS,
00033     GDCM_LESSEOREQUAL
00034 };
00035 enum LodModeType
00036 {
00037     LD_ALL = 0x00000000,
00038     LD_NOSEQ = 0x00000001,
00039     LD_NOSHADOW = 0x00000002,
00040     LD_NOSHADOWSEQ = 0x00000004
00041 };
00042
00043
00044
00045
00046
00047
00048
00049
00050 class GDCM_EXPORT FileWithName : public File
00051 {
00052 public:
00053     FileWithName(File &f):File(f),filename(){}
00054     std::string filename;
00055 };
00056
00057 typedef std::vector< SmartPointer<FileWithName> > FileList;
00058 typedef bool (*BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *);
00059 class Scanner;
00060
00061 class GDCM_EXPORT SerieHelper
00062 {
00063 public:
00064     SerieHelper();
00065     ~SerieHelper();
00066
00067     void Clear();
00068     void SetLoadMode(int ) {};
00069     void SetDirectory(std::string const &dir, bool recursive=false);
00070
00071     void AddRestriction(const std::string & tag);
00072     void SetUseSeriesDetails( bool useSeriesDetails );
00073     void CreateDefaultUniqueSeriesIdentifier();
00074     FileList *GetFirstSingleSerieUIDFileSet();
00075     FileList *GetNextSingleSerieUIDFileSet();
00076     std::string CreateUniqueSeriesIdentifier( File * inFile );
00077     void OrderFileList(FileList *fileSet);
00078     void AddRestriction(uint16_t group, uint16_t elem, std::string const &value, int op);
00079
00080

```

```

00087 protected:
00088     bool UserOrdering(FileList *fileSet);
00089     void AddFileName(std::string const &filename);
00090     bool AddFile(FileWithName &header);
00091     void AddRestriction(const Tag& tag);
00092     bool ImagePositionPatientOrdering(FileList *fileSet);
00093     bool ImageNumberOrdering( FileList *fileList );
00094     bool FileNameOrdering( FileList *fileList );
00095
00096     using Rule = struct RuleStructure{
00097         uint16_t group;
00098         uint16_t elem;
00099         std::string value;
00100         int op;
00101     };
00102     typedef std::vector<Rule> SerieRestrictions;
00103
00104     typedef std::map<std::string, FileList *> SingleSerieUIDFileSetmap;
00105     SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT;
00106     SingleSerieUIDFileSetmap::iterator ItFileSetHt;
00107
00108 private:
00109     SerieRestrictions Restrictions;
00110     SerieRestrictions Refine;
00111
00112     bool UseSeriesDetails;
00113     bool DirectOrder;
00114
00115     BOOL_FUNCTION_PFILE_PFILE_POINTER UserLessThanFunction;
00116 };
00117
00118 // backward compat
00119 } // end namespace gdcm
00120
00121
00122 #endif //GDCMSERIEHELPER_H

```

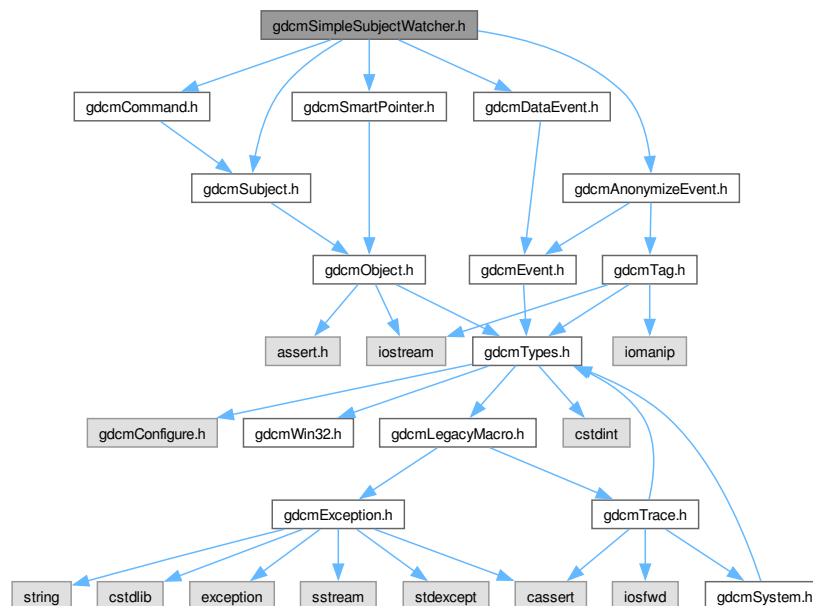
13.413 gdcmSimpleSubjectWatcher.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

```

Include dependency graph for gdcmsimplesubjectwatcher.h:



Classes

- class [gdcmsimplesubjectwatcher::SimpleSubjectWatcher](#)
[SimpleSubjectWatcher](#).

Namespaces

- namespace [gdcmsimplesubjectwatcher](#)

13.414 gdcmsimplesubjectwatcher.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSIMPLESUBJECTWATCHER_H

```

```

00015 #define GDCMSIMPLESUBJECTWATCHER_H
00016
00017 #include "gdcmSubject.h"
00018 #include "gdcmCommand.h"
00019 #include "gdcmSmartPointer.h"
00020 #include "gdcmAnonymizeEvent.h"
00021 #include "gdcmDataEvent.h"
00022
00023 namespace gdcm
00024 {
00025 //-----
00026 class Event;
00031 class GDCM_EXPORT SimpleSubjectWatcher
00032 {
00033 public:
00034 SimpleSubjectWatcher(Subject * s, const char *comment = "");
00035 virtual ~SimpleSubjectWatcher();
00036 SimpleSubjectWatcher(const SimpleSubjectWatcher&) = delete;
00037 void operator=(const SimpleSubjectWatcher&) = delete;
00038
00039 protected:
00040 virtual void StartFilter();
00041 virtual void EndFilter();
00042 virtual void ShowProgress(Subject *caller, const Event &evt);
00043 virtual void ShowFileName(Subject *caller, const Event &evt);
00044 virtual void ShowIteration();
00045 virtual void ShowAnonymization(Subject *caller, const Event &evt);
00046 virtual void ShowDataSet(Subject *caller, const Event &evt);
00047 virtual void ShowData(Subject *caller, const Event &evt);
00048 virtual void ShowAbort();
00049
00050 protected:
00051 // Custom API used for internal Testing do not use !
00052 void TestAbortOn();
00053 void TestAbortOff();
00054
00055 private:
00056 SmartPointer<Subject> m_Subject;
00057 std::string m_Comment;
00058
00059 typedef SimpleMemberCommand<SimpleSubjectWatcher> SimpleCommandType;
00060 typedef MemberCommand<SimpleSubjectWatcher> CommandType;
00061
00062 SmartPointer<SimpleCommandType> m_StartFilterCommand;
00063 SmartPointer<SimpleCommandType> m_EndFilterCommand;
00064 SmartPointer<CommandType> m_ProgressFilterCommand;
00065 SmartPointer<CommandType> m_FileNameFilterCommand;
00066 SmartPointer<SimpleCommandType> m_IterationFilterCommand;
00067 SmartPointer<SimpleCommandType> m_AbortFilterCommand;
00068 SmartPointer<CommandType> m_AnonymizeFilterCommand;
00069 SmartPointer<CommandType> m_DataFilterCommand;
00070 SmartPointer<CommandType> m_DataSetFilterCommand;
00071
00072 unsigned long m_StartTag;
00073 unsigned long m_EndTag;
00074 unsigned long m_ProgressTag;
00075 unsigned long m_FileNameTag;
00076 unsigned long m_IterationTag;
00077 unsigned long m_AbortTag;
00078 unsigned long m_AnonymizeTag;
00079 unsigned long m_DataTag;
00080 unsigned long m_DataSetTag;
00081
00082 bool m_TestAbort;
00083
00084 };
00085 } // end namespace gdcm
00086 //-----
00087 #endif //GDCMSIMPLESUBJECTWATCHER_H

```

13.415 gdcmSorter.h File Reference

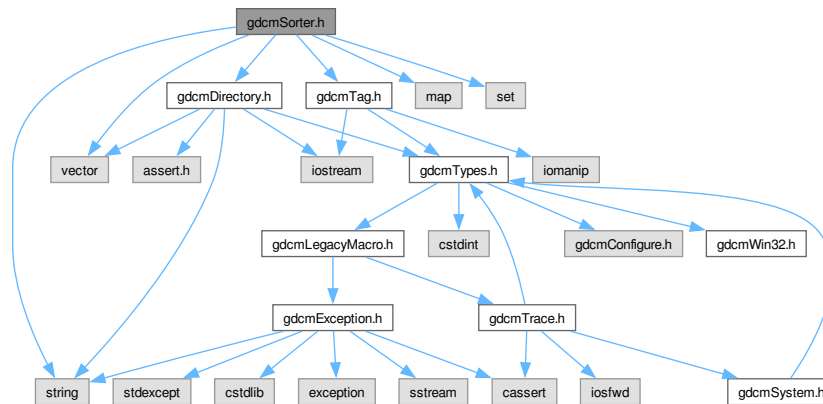
```

#include "gdcmDirectory.h"
#include "gdcmTag.h"

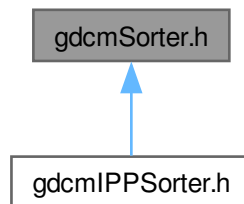
```

```
#include <vector>
#include <string>
#include <map>
#include <set>
```

Include dependency graph for gdcmsorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcmsorter::Sorter](#)
[Sorter](#).

Namespaces

- namespace [gdcmsorter](#)

Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Sorter &s)`

13.416 gdcmmSorter.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSORTER_H
00015 #define GDCMSORTER_H
00016
00017 #include "gdcmmDirectory.h"
00018 #include "gdcmmTag.h"
00019
00020 #include <vector>
00021 #include <string>
00022 #include <map>
00023 #include <set>
00024
00025 namespace gdcmm
00026 {
00027     class DataSet;
00028
00029     class GDCM_EXPORT Sorter
00030     {
00031     friend std::ostream& operator<<(std::ostream &_os, const Sorter &s);
00032     public:
00033         Sorter();
00034         virtual ~Sorter();
00035
00036         virtual bool Sort(std::vector<std::string> & filenames);
00037
00038         const std::vector<std::string> &GetFilenames() const { return Filenames; }
00039
00040         void Print(std::ostream &os) const;
00041
00042         bool AddSelect( Tag const &tag, const char *value );
00043
00044         void SetTagsToRead( std::set<Tag> const & tags );
00045
00046         typedef bool (*SortFunction)(DataSet const &, DataSet const &);
00047         void SetSortFunction( SortFunction f );
00048
00049         virtual bool StableSort(std::vector<std::string> const & filenames);
00050
00051     protected:
00052         std::vector<std::string> Filenames;
00053         typedef std::map<Tag,std::string> SelectionMap;
00054         std::map<Tag,std::string> Selection;
00055         SortFunction SortFunc;
00056         std::set<Tag> TagsToRead;
00057     };
00058 //-----
00059 inline std::ostream& operator<<(std::ostream &os, const Sorter &s)
00060 {
00061     s.Print( os );
00062     return os;
00063 }
00064
00065

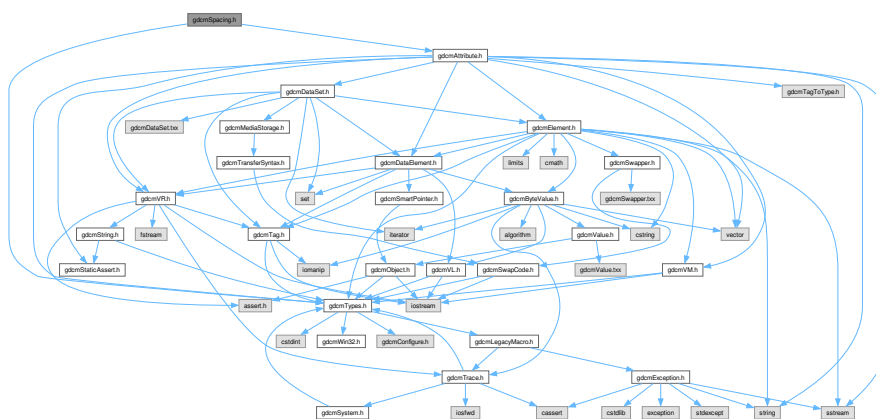
```



```
00085 } // end namespace gdcm
00086
00087 #endif //GDCMSORTER_H
```

13.417 gdcmSpacing.h File Reference

```
#include "gdcTypes.h"
#include "gdcAttribute.h"
Include dependency graph for gdcSpacing.h:
```



Classes

- class `gdc::Spacing`
Class for `Spacing`.

Namespaces

- namespace `gdcm`

13.418 gdcmspacing.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012

```

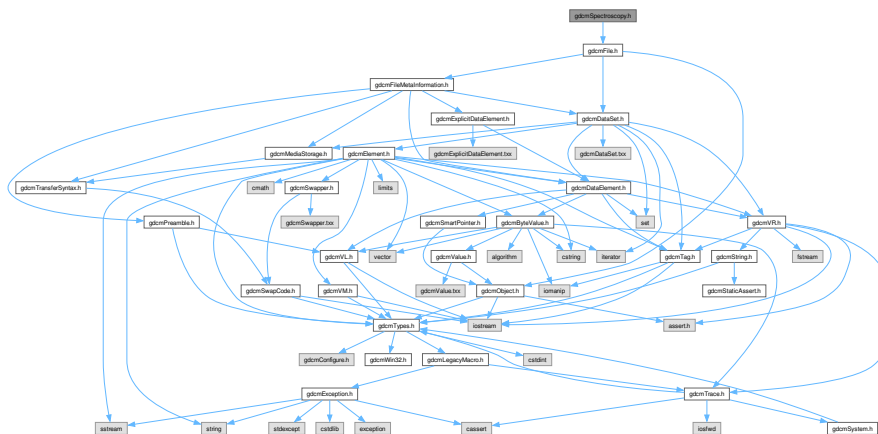
```

00013
===== */
00014 #ifndef GDCMSPACING_H
00015 #define GDCMSPACING_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmAttribute.h"
00019
00020 namespace gdcm
00021 {
00022
00056
00085 class GDCM_EXPORT Spacing
00086 {
00087 public :
00088     Spacing();
00089     ~Spacing() = default;
00090
00091     // Here are the list of spacing we support:
00092     // (0018,0088) DS [1.500000] # 8,1 Spacing Between Slices
00093     // (0018,1164) DS [0.5\0.5 ] # 8,2 Imager Pixel Spacing
00094     // (0018,2010) DS [0.664062\0.664062 ] # 18,2 Nominal Scanned Pixel Spacing
00095     // (0018,7022) DS [0.125\0.125 ] # 12,2 Detector Element Spacing
00096     // (0028,0030) DS [0.25\0.25 ] # 10,2 Pixel Spacing
00097     // > (0028,0a02) CS [FIDUCIAL] # 8,1 Pixel Spacing Calibration Type
00098     // > (0028,0a04) LO [Used fiducial ] # 14,1 Pixel Spacing Calibration Description
00099     // (0028,0034) IS [4\3 ] # 4,2 Pixel Aspect Ratio
00100     // (3002,0011) DS [0.8\0.8 ] # 8,2 Image Plane Pixel Spacing
00101
00102     // Here is the list of Spacing we do not support:
00103     // <entry group="0018" element="7041" vr="LT" vm="1" name="Grid Spacing Material"/>
00104     // <entry group="0018" element="9030" vr="FD" vm="1" name="Tag Spacing First Dimension"/>
00105     // <entry group="0018" element="9218" vr="FD" vm="1" name="Tag Spacing Second Dimension"/>
00106     // <entry group="0018" element="9322" vr="FD" vm="2" name="Reconstruction Pixel Spacing"/>
00107     // <entry group="0018" element="9404" vr="FL" vm="2" name="Object Pixel Spacing in Center of Beam"/>
00108     // <entry group="0040" element="08d8" vr="SQ" vm="1" name="Pixel Spacing Sequence"/>
00109     // <entry group="0070" element="0101" vr="DS" vm="2" name="Presentation Pixel Spacing"/>
00110     // <entry group="2010" element="0376" vr="DS" vm="2" name="Printer Pixel Spacing"/>
00111     // <entry group="300a" element="00e9" vr="DS" vm="2" name="Compensator Pixel Spacing"/>
00112
00113     typedef enum {
00114         DETECTOR = 0, // (0018,1164) Imager Pixel Spacing
00115         MAGNIFIED, // (0018,1114) (IHE Mammo)
00116         CALIBRATED, // (0028,0030) Pixel Spacing -> (0028,0a04) Pixel Spacing Calibration Description
00117         UNKNOWN
00118     } SpacingType;
00119
00120     static Attribute<0x28,0x34> ComputePixelAspectRatioFromPixelSpacing(const Attribute<0x28,0x30>& pixelspacing);
00121 };
00122 } // end namespace gdcm
00123 //-----
00124 #endif //GDCMSPACING_H

```

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmSpectroscopy.h:



- class `gdc::Spectroscopy`
Spectroscopy class.

- namespace `gdcm`

[Go to the documentation of this file.](#)

```
00001 /*=====
00002
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSPECTROSCOPY_H
00015 #define GDCMSPECTROSCOPY_H
00016
00017 #include "gdcmFile.h"
00018
00019 namespace gdcm
```

```

00020 {
00024 class GDCM_EXPORT Spectroscopy
00025 {
00026 public:
00027     Spectroscopy() = default;
00028
00029 private:
00030 };
00031
00032 } // end namespace gdcms
00033
00034 #endif //GDCMSPECTROSCOPY_H

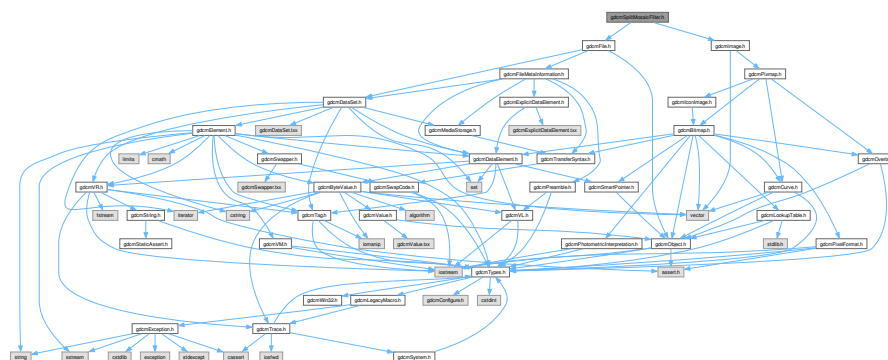
```

13.421 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
```

```
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class [gdcm::SplitMosaicFilter](#)
SplitMosaicFilter class.

Namespaces

- namespace [gdcm](#)

13.422 gdcmSplitMosaicFilter.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004 Copyright (c) 2006-2011 Mathieu Malaterre
00005 All rights reserved.

```

```

00007 See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013 =====*/
00014 #ifndef GDCMSPLITMOSAICFILTER_H
00015 #define GDCMSPLITMOSAICFILTER_H
00016
00017 #include "gdcmsFile.h"
00018 #include "gdcmsImage.h"
00019
00020 namespace gdcms
00021 {
00022
00023 /*
00024 * Everything done in this code is for the sole purpose of writing interoperable
00025 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00026 * If you believe anything in this code violates any law or any of your rights,
00027 * please contact us (gdcms-developers@lists.sourceforge.net) so that we can
00028 * find a solution.
00029 */
00030
00031 class GDCM_EXPORT SplitMosaicFilter
00032 {
00033 public:
00034     SplitMosaicFilter();
00035     ~SplitMosaicFilter();
00036
00037     bool Split();
00038
00039     bool ComputeMOSAICDimensions(unsigned int dims[3]);
00040
00041     bool ComputeMOSAICSliceNormal( double dims[3], bool & inverted );
00042
00043     bool ComputeMOSAICSlicePosition( double pos[3], bool inverted );
00044
00045     bool ComputeMOSAICImagePositionPatient( double pos[3],
00046         const double ipp[6],
00047         const double dircos[6],
00048         const double pixelspacing[3],
00049         const unsigned int image_dims[3] ,
00050         const unsigned int mosaic_dims[3], bool inverted );
00051
00052     void SetImage(const Image& image);
00053     const Image &GetImage() const { return *I; }
00054     Image &GetImage() { return *I; }
00055
00056     void SetFile(const File& f) { F = f; }
00057     File &GetFile() { return *F; }
00058     const File &GetFile() const { return *F; }
00059
00060     static bool GetAcquisitionSize(unsigned int size[2], DataSet const & ds);
00061
00062     static unsigned int GetNumberOfImagesInMosaic( File const & file );
00063
00064     static const DataElement& ComputeCSAImageHeaderInfo(const DataSet& ds, bool handleMissingPrivateCreator = true);
00065
00066     static const DataElement& ComputeCSASeriesHeaderInfo(const DataSet& ds, bool handleMissingPrivateCreator = true);
00067
00068 protected:
00069
00070 private:
00071     SmartPointer<File> F;
00072     SmartPointer<Image> I;
00073 };
00074
00075 } // end namespace gdcms
00076
00077 #endif //GDCMSPLITMOSAICFILTER_H

```



```

00018 #ifndef GDCMSTREAMIMAGEREADER_H
00019 #define GDCMSTREAMIMAGEREADER_H
00020
00021 #include "gdcmReader.h"
00022
00023 namespace gdcm
00024 {
00025
00026 class MediaStorage;
00027 class GDCM__EXPORT StreamImageReader
00028 {
00029 {
00030
00031 public:
00032 StreamImageReader();
00033 virtual ~StreamImageReader();
00034
00035 void SetFileName(const char* inFileName);
00036 void SetStream(std::istream& inStream);
00037
00038 std::vector<unsigned int> GetDimensionsValueForResolution( unsigned int );
00039
00040 void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
00041   uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
00042
00043 uint32_t DefineProperBufferLength() const;
00044
00045 bool Read(char* inReadBuffer, const std::size_t& inBufferLength);
00046
00047 bool CanReadImage() const;
00048
00049 virtual bool ReadImageInformation();
00050
00051 File const & GetFile() const;
00052
00053 protected:
00054 private:
00055 //contains a reader for being able to ReadUpToTag
00056 //however, we don't want the user to be able to call Read
00057 //either directly or via a parent class call, so we hide the reader in here.
00058 Reader mReader;
00059
00060 std::streamoff mFileOffset; //the file offset for getting header information
00061 #if 0
00062 std::streamoff mFileOffset1;
00063 #endif
00064 DataSet mHeaderInformation; //all the non-pixel information
00065
00066 //for thread safety, these should not be stored here, but should be used
00067 //for every read subregion operation.
00068 uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;
00069
00070 bool ReadImageSubregionRAW(char* inReadBuffer, const std::size_t& inBufferLength);
00071
00072 bool ReadImageSubregionJpegLS(char* inReadBuffer, const std::size_t& inBufferLength);
00073 };
00074 };
00075 // end namespace gdcm
00076 #endif //GDCMSTREAMIMAGEREADER_H
00077

```

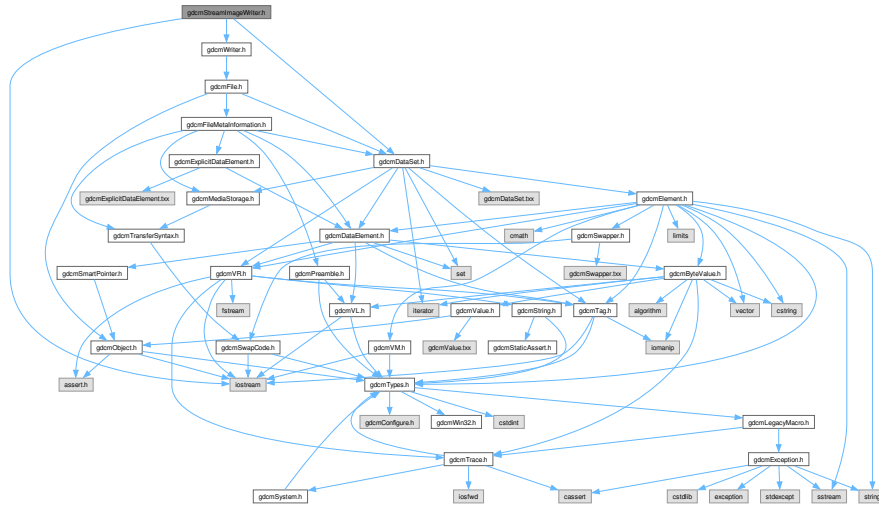
13.425 gdcmStreamImageWriter.h File Reference

```

#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"

```

Include dependency graph for `gdcmStreamImageWriter.h`:



Classes

- class `gdcm::StreamImageWriter`
`StreamImageReader`.

Namespaces

- namespace `gdcm`

13.426 gdcmStreamImageWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018
00019 #ifndef GDCMSTREAMIMAGEWRITER_H
00020 #define GDCMSTREAMIMAGEWRITER_H
00021
00022 #include "gdcmWriter.h"

```



```

00023 #include <iostream>
00024 #include "gdcmDataSet.h"
00025
00026 namespace gdcm
00027 {
00028
00029 class MediaStorage;
00030 class RAWCodec;
00042 class GDCM_EXPORT StreamImageWriter
00043 {
00044
00045 public:
00046     StreamImageWriter();
00047     virtual ~StreamImageWriter();
00048
00049
00053 void SetFileName(const char* inFileName);
00054 void SetStream(std::ostream& inStream);
00055
00064 void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
00065     uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);
00066
00067
00073 uint32_t DefineProperBufferLength();
00074
00082 bool Write(void* inWriteBuffer, const std::size_t& inBufferLength);
00083
00087 virtual bool WriteImageInformation();
00088
00092 bool CanWriteFile() const;
00093
00094
00097 void SetFile(const File& inFile);
00098
00099 protected:
00100
00101 //contains the PrepareWrite function, which will get the given dataset ready
00102 //for writing to disk by manufacturing the header information.
00103 //note that if there is a pixel element in the given dataset, that will be removed
00104 //during the copy, so that the imagewriter can write everything else out
00105 Writer mWriter;
00106
00107 //is the offset necessary if we always append?
00108 //std::streamoff mFileOffset; //the fileoffset for getting header information
00109 SmartPointer<File> mspFile; //all the non-pixel information
00110
00111 //for thread safety, these should not be stored here, but should be used
00112 //for every read subregion operation.
00113 uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;
00114
00119 //virtual bool ReadImageSubregionRAW(std::ostream& os);
00120 virtual bool WriteImageSubregionRAW(char* inWriteBuffer, const std::size_t& inBufferLength);
00121
00131 int WriteRawHeader(RAWCodec* inCodec, std::ostream* inStream);
00132
00137 int mElementOffsets;
00138 int mElementOffsets1;
00139
00140 };
00141
00142
00143 } // end namespace gdcm
00144
00145 #endif //GDCMSTREAMIMAGEWRITER_H

```

13.427 gdcmStrictScanner.h File Reference

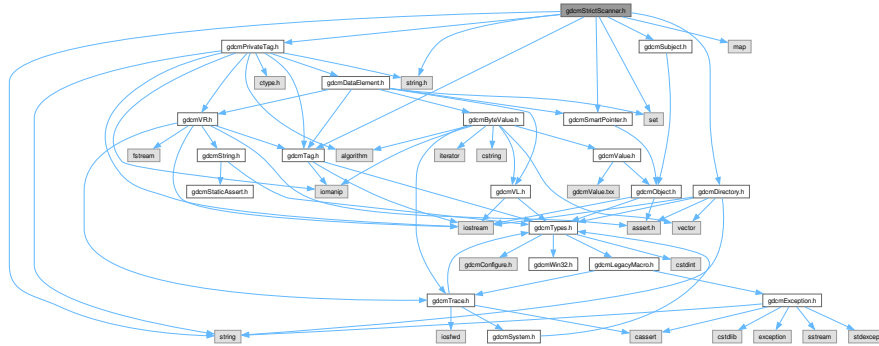
```

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>

```

```
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for `gdcmStrictScanner.h`:



Classes

- struct `gdcm::StrictScanner::ltstr`
- class `gdcm::StrictScanner`
[StrictScanner](#).

Namespaces

- namespace `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner &s)`

13.428 gdcmStrictScanner.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSTRICTSCANNER_H
00015 #define GDCMSTRICTSCANNER_H
```

```

00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmSubject.h"
00019 #include "gdcmTag.h"
00020 #include "gdcmPrivateTag.h"
00021 #include "gdcmSmartPointer.h"
00022
00023 #include <map>
00024 #include <set>
00025 #include <string>
00026
00027 #include <string.h> // strcmp
00028
00029 namespace gdcm
00030 {
00031 class StringFilter;
00032
00033 class GDCM_EXPORT StrictScanner : public Subject
00034 {
00035 friend std::ostream& operator<<(std::ostream &_os, const StrictScanner &s);
00036 public:
00037 StrictScanner():Values(),FileNames(),Mappings(),Progress(0.0) {}
00038 ~StrictScanner() override;
00039
00040 typedef std::map<Tag, const char*> TagToValue;
00041 //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
00042 //typedef TagToStringMap TagToValue;
00043 typedef TagToValue::value_type TagToValueValueType;
00044
00045 void AddTag( Tag const & t );
00046 void ClearTags();
00047
00048 // Work in progress do not use:
00049 void AddPrivateTag( PrivateTag const & t );
00050
00051 void AddSkipTag( Tag const & t );
00052 void ClearSkipTags();
00053
00054 bool Scan( Directory::FileNamesType const & filenames );
00055
00056 Directory::FileNamesType const &GetFileNames() const { return FileNames; }
00057
00058 void Print( std::ostream & os ) const override;
00059
00060 void PrintTable( std::ostream & os ) const;
00061
00062 bool IsKey( const char * filename ) const;
00063
00064 Directory::FileNamesType GetKeys() const;
00065
00066 // struct to store all the values found:
00067 typedef std::set< std::string > ValuesType;
00068
00069 ValuesType const & GetValues() const { return Values; }
00070
00071 ValuesType GetValues(Tag const &t) const;
00072
00073 Directory::FileNamesType GetOrderedValues(Tag const &t) const;
00074
00075 /* Itstr is CRITICAL, otherwise pointers value are used to do the key comparison */
00076 struct Itstr
00077 {
00078     bool operator()(const char* s1, const char* s2) const
00079     {
00080         {
00081             gdcm_assert( s1 && s2 );
00082             return strcmp(s1, s2) < 0;
00083         }
00084     };
00085     typedef std::map<const char *,TagToValue, Itstr> MappingType;
00086     typedef MappingType::const_iterator ConstIterator;
00087     ConstIterator Begin() const { return Mappings.begin(); }
00088     ConstIterator End() const { return Mappings.end(); }
00089
00090 MappingType const & GetMappings() const { return Mappings; }
00091
00092 TagToValue const & GetMapping(const char *filename) const;
00093
00094 const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;
00095
00096 Directory::FileNamesType GetAllFileNamesFromTagToValue(Tag const &t, const char *valueref) const;
00097
00098

```


Classes

- struct [gdcm::StrictScanner2::ltstr](#)
- class [gdcm::StrictScanner2](#)
[StrictScanner2](#).

Namespaces

- namespace [gdcm](#)

Functions

- [std::ostream & gdcm::operator<< \(std::ostream &os, const StrictScanner2 &s\)](#)

13.430 gdcmStrictScanner2.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSTRICTSCANNER2_H
00015  #define GDCMSTRICTSCANNER2_H
00016
00017  #include "gdcmDirectory.h"
00018  #include "gdcmPrivateTag.h"
00019  #include "gdcmSmartPointer.h"
00020  #include "gdcmSubject.h"
00021  #include "gdcmTag.h"
00022
00023  #include <map>
00024  #include <set>
00025  #include <string>
00026
00027  #include <string.h> // strcmp
00028
00029  namespace gdcm {
00030  class StringFilter;
00031
00032  class GDCM_EXPORT StrictScanner2 : public Subject {
00033  friend std::ostream &operator<<(std::ostream &_os, const StrictScanner2 &s);
00034
00035  public:
00036  StrictScanner2() : Values(), Filenames(), PublicMappings(), PrivateMappings(), Progress(0.0) {}
00037  ~StrictScanner2() override;
00038
00039  typedef std::map<Tag, const char *> PublicTagToValue;
00040  typedef PublicTagToValue::value_type PublicTagToValueValueType;
00041
00042  typedef std::map<PrivateTag, const char *> PrivateTagToValue;
00043  typedef PrivateTagToValue::value_type PrivateTagToValueValueType;
00044
00045  bool AddPublicTag(Tag const &t);
00046  void ClearPublicTags();

```

```

00074
00075 // Work in progress do not use:
00076 bool AddPrivateTag(PrivateTag const &pt);
00077 void ClearPrivateTags();
00078
00080 bool AddSkipTag(Tag const &t);
00081 void ClearSkipTags();
00082
00084 bool Scan(Directory::FileNamesType const &filenames);
00085
00087 Directory::FileNamesType const &GetFileNames() const { return Filenames; }
00088
00090 void Print(std::ostream &os) const override;
00091
00093 void PrintTable(std::ostream &os, bool header = false) const;
00094
00098 bool IsKey(const char *filename) const;
00099
00102 Directory::FileNamesType GetKeys() const;
00103
00104 // struct to store all the values found:
00105 typedef std::set<std::string> ValuesType;
00106
00108 ValuesType const &GetValues() const { return Values; }
00109
00111 ValuesType GetPublicValues(Tag const &t) const;
00112
00115 ValuesType GetPrivateValues(PrivateTag const &pt) const;
00116
00120 Directory::FileNamesType GetPublicOrderedValues(Tag const &t) const;
00121
00122 Directory::FileNamesType GetPrivateOrderedValues(PrivateTag const &pt) const;
00123
00124 /* Itstr is CRITICAL, otherwise pointers value are used to do the key
00125  * comparison */
00126 struct Itstr {
00127     bool operator()(const char *s1, const char *s2) const {
00128         gdcmm_assert(s1 && s2);
00129         return strcmp(s1, s2) < 0;
00130     }
00131 };
00132 typedef std::map<const char *, PublicTagToValue, Itstr> PublicMappingType;
00133 typedef PublicMappingType::const_iterator PublicConstIterator;
00134 PublicConstIterator Begin() const { return PublicMappings.begin(); }
00135 PublicConstIterator End() const { return PublicMappings.end(); }
00136
00137 typedef std::map<const char *, PrivateTagToValue, Itstr> PrivateMappingType;
00138 typedef PrivateMappingType::const_iterator PrivateConstIterator;
00139 PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
00140 PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }
00141
00144 PublicMappingType const &GetPublicMappings() const { return PublicMappings; }
00145 PrivateMappingType const &GetPrivateMappings() const {
00146     return PrivateMappings;
00147 }
00148
00150 PublicTagToValue const &GetPublicMapping(const char *filename) const;
00151 PrivateTagToValue const &GetPrivateMapping(const char *filename) const;
00152
00155 const char *GetFilenameFromPublicTagToValue(Tag const &t,
00156                                             const char *valueref) const;
00157 const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt,
00158                                             const char *valueref) const;
00159
00162 Directory::FileNamesType GetAllFileNamesFromPublicTagToValue(
00163     Tag const &t, const char *valueref) const;
00164 Directory::FileNamesType GetAllFileNamesFromPrivateTagToValue(
00165     PrivateTag const &pt, const char *valueref) const;
00166
00169 // by a call to GetMapping()
00170 PublicTagToValue const &GetMappingFromPublicTagToValue(
00171     Tag const &t, const char *value) const;
00172 PrivateTagToValue const &GetMappingFromPrivateTagToValue(
00173     PrivateTag const &pt, const char *value) const;
00174
00180 const char *GetPublicValue(const char *filename, Tag const &t) const;
00181 const char *GetPrivateValue(const char *filename, PrivateTag const &t) const;
00182
00184 static SmartPointer<StrictScanner2> New() { return new StrictScanner2; }
00185
00186 protected:

```

```

00187 void ProcessPublicTag(StringFilter &sf, const char *filename);
00188 void ProcessPrivateTag(StringFilter &sf, const char *filename);
00189
00190 private:
00191 // struct to store all uniq tags in ascending order:
00192 typedef std::set<Tag> PublicTagsType;
00193 typedef std::set<PrivateTag> PrivateTagsType;
00194 std::set<Tag> PublicTags; // Public and Private Creator
00195 std::set<PrivateTag> PrivateTags; // Only Private (no Private Creator)
00196 std::set<Tag> SkipTags;
00197 ValueType Values;
00198 Directory::FileNamesType Filenames;
00199
00200 // Main struct that will hold all public mapping:
00201 PublicMappingType PublicMappings;
00202 // Main struct that will hold all private mapping:
00203 PrivateMappingType PrivateMappings;
00204
00205 double Progress;
00206 };
00207 //-----
00208 inline std::ostream &operator<<(std::ostream &os, const StrictScanner2 &s) {
00209     s.Print(os);
00210     return os;
00211 }
00212
00213 } // end namespace gdc
00214
00215 #endif // GDCMSTRICTSCANNER2_H

```

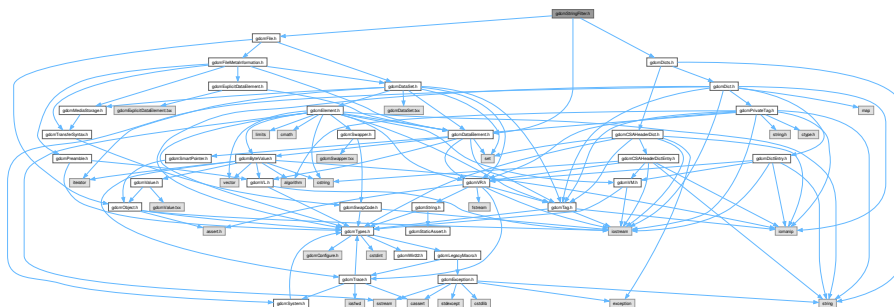
13.431 gdcStringFilter.h File Reference

#include "gdcmDataElement.h"

#include "gdcmDicts.h"

#include "gdcmFile.h"

Include dependency graph for gdcStringFilter.h:



Classes

- class [gdcm::StringFilter](#)
[StringFilter](#).

Namespaces

- namespace [gdcm](#)

13.432 gdcmStringFilter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMSTRINGFILTER_H
00015  #define GDCMSTRINGFILTER_H
00016  00017  #include "gdcmDataElement.h"
00018  #include "gdcmDicts.h"
00019  #include "gdcmFile.h"
00020  00021  namespace gdcm
00022  {
00023  00029  class GDCM_EXPORT StringFilter
00030  {
00031  public:
00032  StringFilter();
00033  ~StringFilter();
00034  00036  void UseDictAlways(bool) {}
00037  00039  void SetDicts(const Dicts &dicts);
00040  00044  std::string ToString(const DataElement& de) const;
00045  00047  std::string ToString(const Tag& t) const;
00048  00049  std::string ToString(const PrivateTag& t) const;
00050  00055  std::pair<std::string, std::string> ToStringPair(const DataElement& de) const;
00057  std::pair<std::string, std::string> ToStringPair(const Tag& t) const;
00058  00060  std::string FromString(const Tag&t, const char * value, size_t len);
00061  00063  void SetFile(const File& f) { F = f; }
00064  File &GetFile() { return *F; }
00065  const File &GetFile() const { return *F; }
00066  00070  bool ExecuteQuery(std::string const &query, std::string & value) const;
00071  00072  protected:
00073  std::pair<std::string, std::string> ToStringPair(const Tag& t, DataSet const &ds) const;
00074  bool ExecuteQuery(std::string const &query, DataSet const &ds, std::string & value) const;
00075  00076  private:
00077  std::pair<std::string, std::string> ToStringPairInternal(const DataElement& de, DataSet const &ds) const;
00078  SmartPointer<File> F;
00079  };
00080  00081  } // end namespace gdcm
00082  00083  #endif //GDCMSTRINGFILTER_H

```

13.433 gdcmSurface.h File Reference

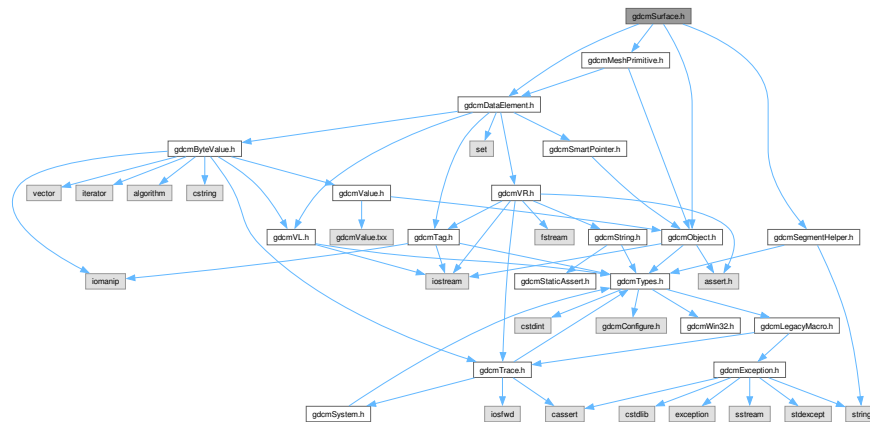
```

#include <gdcmObject.h>
#include <gdcmDataElement.h>

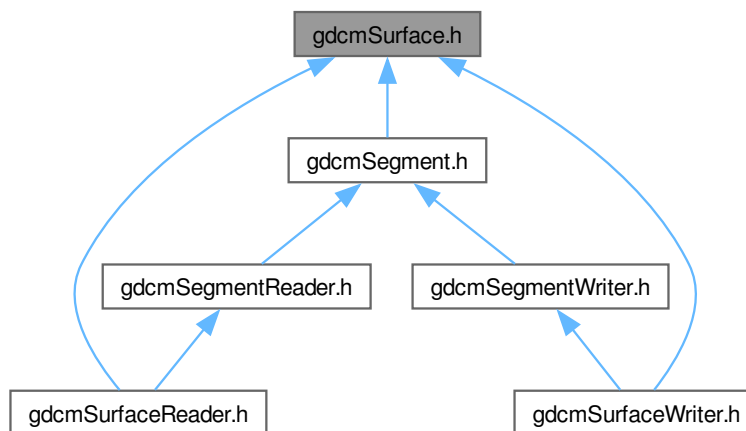
```



```
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"
Include dependency graph for gdcmSurface.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::Surface`
This class defines a SURFACE IE.

Namespaces

- namespace `gdcm`

13.434 gdcmSurface.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSURFACE_H
00015 #define GDCMSURFACE_H
00016
00017 #include <gdcmObject.h>
00018 #include <gdcmDataElement.h>
00019 #include <gdcmMeshPrimitive.h>
00020 #include "gdcmSegmentHelper.h" // for BasicCodedEntry
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT Surface : public Object
00026 {
00027 public:
00028
00029 typedef enum {
00030     NO = 0,
00031     YES,
00032     UNKNOWN,
00033     STATES_END
00034 } STATES;
00035
00036 static const char * GetSTATESString(STATES state);
00037 static STATES GetSTATES(const char * state);
00038
00039 typedef enum {
00040     SURFACE = 0,
00041     WIREFRAME,
00042     POINTS,
00043     VIEWType_END
00044 } VIEWType;
00045
00046 static const char * GetVIEWTypeString(VIEWType type);
00047 static VIEWType GetVIEWType(const char * type);
00048
00049 Surface();
00050 ~Surface() override;
00051
00052 /** Common getters/setters */
00053 unsigned long GetSurfaceNumber() const;
00054 void SetSurfaceNumber(const unsigned long nb);
00055
00056 const char * GetSurfaceComments() const;
00057 void SetSurfaceComments(const char * comment);
00058
00059 bool GetSurfaceProcessing() const;
00060 void SetSurfaceProcessing(bool b);
00061
00062 float GetSurfaceProcessingRatio() const;
00063 void SetSurfaceProcessingRatio(const float ratio);
00064
00065 const char * GetSurfaceProcessingDescription() const;
00066 void SetSurfaceProcessingDescription(const char * description);
00067
00068 SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm() const;
00069 SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm();
00070 void SetProcessingAlgorithm(SegmentHelper::BasicCodedEntry const & BSE);
00071
00072 unsigned short GetRecommendedDisplayGrayscaleValue() const;

```

```

00085 void SetRecommendedDisplayGrayscaleValue(const unsigned short vl);
00086
00087 const unsigned short * GetRecommendedDisplayCIELabValue() const;
00088 unsigned short GetRecommendedDisplayCIELabValue(const unsigned int idx) const;
00089 void SetRecommendedDisplayCIELabValue(const unsigned short vl[3]);
00090 void SetRecommendedDisplayCIELabValue(const unsigned short vl, const unsigned int idx = 0);
00091 void SetRecommendedDisplayCIELabValue(const std::vector< unsigned short > & vl);
00092
00093 float GetRecommendedPresentationOpacity() const;
00094 void SetRecommendedPresentationOpacity(const float opacity);
00095
00096 VIEWType GetRecommendedPresentationType() const;
00097 void SetRecommendedPresentationType(VIEWType type);
00098
00099 STATES GetFiniteVolume() const;
00100 void SetFiniteVolume(STATES state);
00101
00102 STATES GetManifold() const;
00103 void SetManifold(STATES state);
00104
00105 SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily() const;
00106 SegmentHelper::BasicCodedEntry & GetAlgorithmFamily();
00107 void SetAlgorithmFamily(SegmentHelper::BasicCodedEntry const & BSE);
00108
00109 const char * GetAlgorithmVersion() const;
00110 void SetAlgorithmVersion(const char * str);
00111
00112 const char * GetAlgorithmName() const;
00113 void SetAlgorithmName(const char * str);
00114
00115 /** Points getters/setters */
00116 unsigned long GetNumberOfSurfacePoints() const;
00117 void SetNumberOfSurfacePoints(const unsigned long nb);
00118
00119 const DataElement & GetPointCoordinatesData() const;
00120 DataElement & GetPointCoordinatesData();
00121
00122 void SetPointCoordinatesData(DataElement const & de);
00123
00127 const float * GetPointPositionAccuracy() const;
00128 void SetPointPositionAccuracy(const float * accuracies);
00129
00130 float GetMeanPointDistance() const;
00131 void SetMeanPointDistance(float average);
00132
00133 float GetMaximumPointDistance() const;
00134 void SetMaximumPointDistance(float maximum);
00135
00139 const float * GetPointsBoundingBoxCoordinates() const;
00140 void SetPointsBoundingBoxCoordinates(const float * coordinates);
00141
00145 const float * GetAxisOfRotation() const;
00146 void SetAxisOfRotation(const float * axis);
00147
00151 const float * GetCenterOfRotation() const;
00152 void SetCenterOfRotation(const float * center);
00153
00154 /** Vectors getters/setters */
00155 unsigned long GetNumberOfVectors() const;
00156 void SetNumberOfVectors(const unsigned long nb);
00157
00158 unsigned short GetVectorDimensionality() const;
00159 void SetVectorDimensionality(const unsigned short dim);
00160
00161 const float * GetVectorAccuracy() const;
00162 void SetVectorAccuracy(const float * accuracy);
00163
00164 const DataElement & GetVectorCoordinateData() const;
00165 DataElement & GetVectorCoordinateData();
00166
00167 void SetVectorCoordinateData(DataElement const & de);
00168
00169 /** Primitive getters/setters */
00170 MeshPrimitive const & GetMeshPrimitive() const;
00171 MeshPrimitive & GetMeshPrimitive();
00172
00173 void SetMeshPrimitive(MeshPrimitive const & mp);
00174
00175 private:
00176
00177 /** Common members */

```

```

00178
00179 //0066 0003 UL 1 Surface Number
00180 unsigned long SurfaceNumber;
00181 //0066 0004 LT 1 Surface Comments
00182 std::string SurfaceComments;
00183
00184 //0066 0009 CS 1 Surface Processing
00185 bool SurfaceProcessing;
00186 //0066 000a FL 1 Surface Processing Ratio
00187 float SurfaceProcessingRatio;
00188 //0066 000b LO 1 Surface Processing Description
00189 std::string SurfaceProcessingDescription;
00190 // Processing Algorithm Code
00191 SegmentHelper::BasicCodedEntry ProcessingAlgorithm;
00192
00193 //0062 000c US 1 Recommended Display Grayscale Value
00194 unsigned short RecommendedDisplayGrayscaleValue;
00195 //0062 000d US 3 Recommended Display CIELab Value
00196 unsigned short RecommendedDisplayCIELabValue[3];
00197
00198 // 0066 000c FL 1 Recommended Presentation Opacity
00199 float RecommendedPresentationOpacity;
00200 // 0066 000d CS 1 Recommended Presentation Type
00201 VIEWType RecommendedPresentationType;
00202
00203 //0066 000e CS 1 Finite Volume
00204 STATES FiniteVolume;
00205 //0066 0010 CS 1 Manifold
00206 STATES Manifold;
00207
00208 // Algorithm Family Code
00209 SegmentHelper::BasicCodedEntry AlgorithmFamily;
00210
00211 //0066 0031 LO 1 Algorithm Version
00212 std::string AlgorithmVersion;
00213 //0066 0032 LT 1 Algorithm Parameters
00214 //0066 0036 LO 1 Algorithm Name
00215 std::string AlgorithmName;
00216
00217
00218 /** Point members **/
00219
00220 //0066 0015 UL 1 Number of Surface Points
00221 unsigned long NumberOfSurfacePoints;
00222 //0066 0016 OF 1 Point Coordinates Data
00223 DataElement PointCoordinatesData;
00224 //0066 0017 FL 3 Point Position Accuracy
00225 float * PointPositionAccuracy;
00226 //0066 0018 FL 1 Mean Point Distance
00227 float MeanPointDistance;
00228 //0066 0019 FL 1 Maximum Point Distance
00229 float MaximumPointDistance;
00230 //0066 001a FL 6 Points Bounding Box Coordinates
00231 float * PointsBoundingBoxCoordinates;
00232 //0066 001b FL 3 Axis of Rotation
00233 float * AxisOfRotation;
00234 //0066 001c FL 3 Center of Rotation
00235 float * CenterOfRotation;
00236
00237
00238 /** Normal members **/
00239
00240 //0066 001e UL 1 Number of Vectors
00241 unsigned long NumberOfVectors;
00242 //0066 001f US 1 Vector Dimensionality
00243 unsigned short VectorDimensionality;
00244 //0066 0020 FL 1-n Vector Accuracy
00245 float * VectorAccuracy;
00246 //0066 0021 OF 1 Vector Coordinate Data
00247 DataElement VectorCoordinateData;
00248
00249
00250 /** Primitive members **/
00251 SmartPointer< MeshPrimitive > Primitive;
00252 };
00253
00254 }
00255
00256 #endif // GDCMSURFACE_H

```

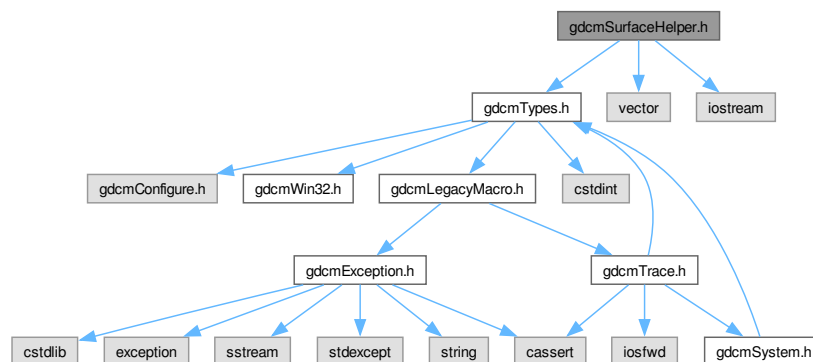
13.435 gdcmSurfaceHelper.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <vector>
```

```
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



Classes

- class [gdcm::SurfaceHelper](#)
[SurfaceHelper](#).

Namespaces

- namespace [gdcm](#)

13.436 gdcmSurfaceHelper.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2017 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSURFACEHELPER_H
00015  #define GDCMSURFACEHELPER_H
00016
00017  #include "gdcmTypes.h" // for GDCM_EXPORT

```

```

00018
00019 #include <vector>
00020 #include <iostream>
00021
00022 namespace gdcm
00023 {
00024
00025 class GDCM_EXPORT SurfaceHelper
00026 {
00027 public:
00028
00029 typedef std::vector< unsigned short > ColorArray;
00030
00031 template <typename T, typename U>
00032 static unsigned short RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
00033                                                         const U rangeMax = 255);
00034
00035 template <typename T, typename U>
00036 static ColorArray RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
00037                                                  const U rangeMax = 255);
00038
00039 template <typename T, typename U>
00040 static std::vector<T> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00041                                                     const U rangeMax = 255);
00042
00043 template <typename U>
00044 static std::vector<float> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00045                                                         const U rangeMax = 255);
00046
00047 private:
00048
00049 static std::vector< float > RGBToXYZ(const std::vector<float> & RGB);
00050
00051 static std::vector< float > XYZToRGB(const std::vector<float> & XYZ);
00052
00053 static std::vector< float > XYZToCIELab(const std::vector<float> & XYZ);
00054
00055 static std::vector< float > CIELabToXYZ(const std::vector<float> & CIELab);
00056 };
00057
00058 template <typename T, typename U>
00059 unsigned short SurfaceHelper::RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
00060                                                                const U rangeMax/* = 255*/ )
00061 {
00062     gdcm_assert(RGB.size() > 2);
00063
00064     unsigned short Grayscale = 0;
00065
00066     const float inverseRangeMax = 1.0f / (float) rangeMax;
00067
00068     // 0xFFFF == 255 == white
00069     Grayscale = (unsigned short) ((0.2989 * RGB[0] + 0.5870 * RGB[1] + 0.1140 * RGB[2])
00070                                 * inverseRangeMax // Convert to range 0-1
00071                                 * 0xFFFF); // Convert to range 0x0000-0xFFFF
00072
00073     return Grayscale;
00074 }
00075
00076 template <typename T, typename U>
00077 SurfaceHelper::ColorArray SurfaceHelper::RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
00078                                                                           const U rangeMax/* = 255*/ )
00079 {
00080     gdcm_assert(RGB.size() > 2);
00081
00082     ColorArray CIELab(3);
00083     std::vector<float> tmp(3);
00084
00085     // Convert to range 0-1
00086     const float inverseRangeMax = 1.0f / (float) rangeMax;
00087     tmp[0] = (float) (RGB[0] * inverseRangeMax);
00088     tmp[1] = (float) (RGB[1] * inverseRangeMax);
00089     tmp[2] = (float) (RGB[2] * inverseRangeMax);
00090
00091     tmp = SurfaceHelper::XYZToCIELab( SurfaceHelper::RGBToXYZ( tmp ) );
00092
00093     // Convert to range 0x0000-0xFFFF
00094     // 0xFFFF == 127, 0x8080 == 0, 0x0000 == -128
00095     CIELab[0] = (unsigned short) ( 0xFFFF * (tmp[0]*0.01f));
00096     if(tmp[1] >= -128 && tmp[1] <= 0)
00097     {
00098         CIELab[1] = (unsigned short)((float)(0x8080)/128.0f)*tmp[1] + ((float)0x8080);
00099     }
00100     else if(tmp[1] <= 127 && tmp[1] > 0)
00101     {

```

```

00146     CIELab[1] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[1] + (float)(0x8080));
00147   }
00148   if(tmp[2] >= -128 && tmp[2] <= 0)
00149   {
00150     CIELab[2] = (unsigned short)(((float)0x8080/128.0f)*tmp[2] + ((float)0x8080));
00151   }
00152   else if(tmp[2] <= 127 && tmp[2] > 0)
00153   {
00154     CIELab[2] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[2] + (float)(0x8080));
00155   }
00156
00157   return CIELab;
00158 }
00159
00160 template <typename T, typename U>
00161 std::vector<T> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00162                                                            const U rangeMax/* = 255 */)
00163 {
00164   gdcm_assert(CIELab.size() > 2);
00165
00166   std::vector<T> RGB(3);
00167   std::vector<float> tmp(3);
00168
00169   // Convert to range 0-1
00170
00171   tmp[0] = 100.0f*CIELab[0] / (float)(0xFFFF);
00172   if(CIELab[1] <= 0x8080)
00173   {
00174     tmp[1] = (float)(((CIELab[1] - 0x8080) * 128.0f) / (float)0x8080);
00175   }
00176   else
00177   {
00178     tmp[1] = (float)((CIELab[1]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
00179   }
00180   if(CIELab[2] <= 0x8080)
00181   {
00182     tmp[2] = (float)(((CIELab[2] - 0x8080) * 128.0f) / (float)0x8080);
00183   }
00184   else
00185   {
00186     tmp[2] = (float)((CIELab[2]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
00187   }
00188
00189   tmp = SurfaceHelper::XYZToRGB( SurfaceHelper::CIELabToXYZ( tmp ) );
00190
00191   // Convert to range 0-rangeMax
00192   RGB[0] = (T) (tmp[0] * rangeMax);
00193   RGB[1] = (T) (tmp[1] * rangeMax);
00194   RGB[2] = (T) (tmp[2] * rangeMax);
00195
00196   return RGB;
00197 }
00198
00199 template <typename U>
00200 std::vector<float> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
00201                                                                const U rangeMax/* = 255 */)
00202 {
00203   return RecommendedDisplayCIELabToRGB<float>(CIELab, rangeMax);
00204 }
00205
00206 } // end namespace gdcm
00207
00208 #endif // GDCMSURFACEHELPER_H

```

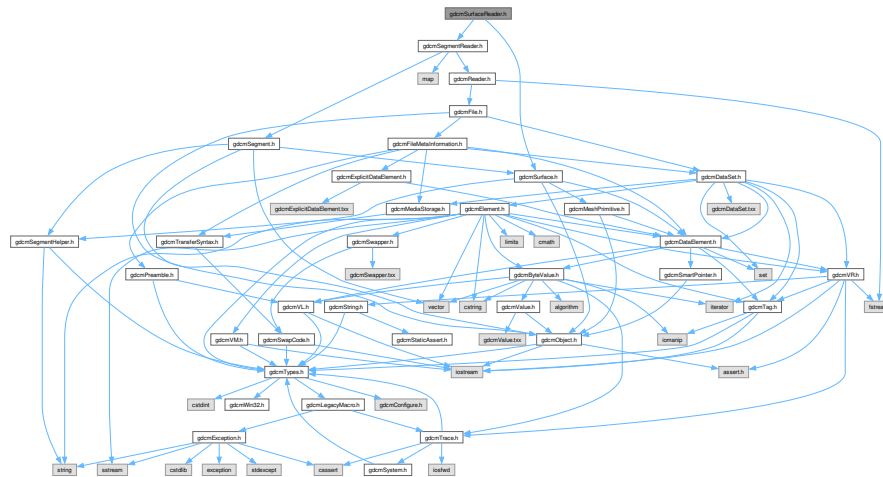
13.437 gdcmSurfaceReader.h File Reference

```

#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>

```

Include dependency graph for `gdcmSurfaceReader.h`:



Classes

- class `gdcm::SurfaceReader`
This class defines a SURFACE IE reader.

Namespaces

- namespace `gdcm`

13.438 gdcmSurfaceReader.h

[Go to the documentation of this file.](#)

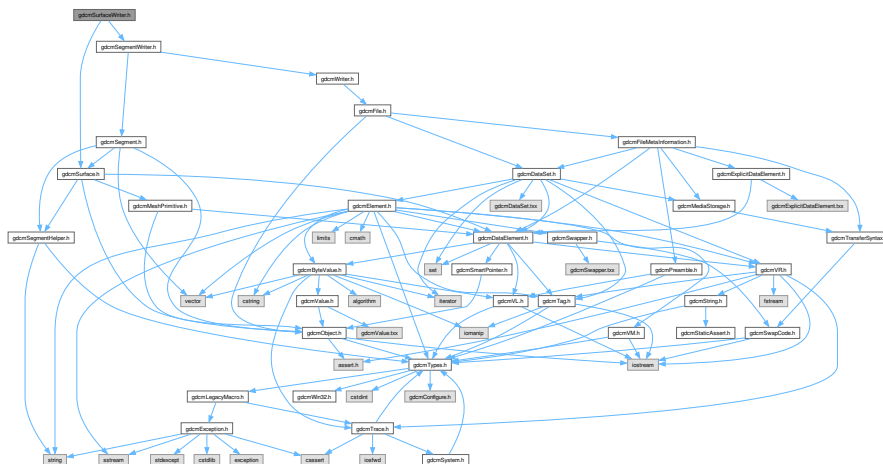
```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  ===== */
00014  #ifndef GDCMSURFACEREADER_H
00015  #define GDCMSURFACEREADER_H
00016
00017  #include <gdcmSegmentReader.h>
00018  #include <gdcmSurface.h>
00019
00020  namespace gdcm
00021  {
00022
00029  class GDCM_EXPORT SurfaceReader : public SegmentReader

```


13.439 gdcmsurfacewriter.h File Reference

Include dependency graph for `gdcmsurfacewriter.h`:



This class defines a SURFACE IE writer.

- namespace `gdcm`

13.440 gdcmSurfaceWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMSURFACEWRITER_H
00015 #define GDCMSURFACEWRITER_H
00016
00017 #include <gdcmSegmentWriter.h>
00018 #include <gdcmSurface.h>
00019
00020 namespace gdcm
00021 {
00022
00023 class GDCM_EXPORT SurfaceWriter : public SegmentWriter
00024 {
00025 public:
00026     SurfaceWriter();
00027     ~SurfaceWriter() override;
00028
00029     // const Surface & GetSurface() const { return *SurfaceData; }
00030     // Surface & GetSurface() { return *SurfaceData; }
00031     // void SetSurface(Surface const & segment);
00032
00033     bool Write() override; // Execute()
00034
00035     unsigned long GetNumberOfSurfaces();
00036     void SetNumberOfSurfaces(const unsigned long nb);
00037
00038 protected:
00039
00040     bool PrepareWrite();
00041
00042     void ComputeNumberOfSurfaces();
00043
00044     bool PrepareWritePointMacro(SmartPointer< Surface > surface,
00045                                 DataSet & surfaceDS,
00046                                 const TransferSyntax & ts);
00047
00048     //0066 0001 UL 1 Number of Surfaces
00049     unsigned long NumberOfSurfaces;
00050 };
00051
00052 }
00053
00054 #endif // GDCMSURFACEWRITER_H

```

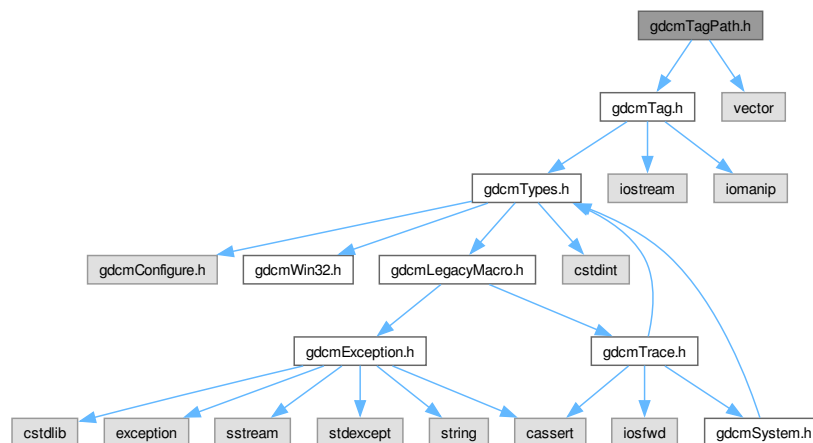
13.441 gdcmTagPath.h File Reference

```

#include "gdcmTag.h"
#include <vector>

```

Include dependency graph for gdcmTagPath.h:



Classes

- class [gdcm::TagPath](#)
class to handle a path of tag.

Namespaces

- namespace [gdcm](#)

13.442 gdcmTagPath.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMTAGPATH_H
00015  #define GDCMTAGPATH_H
00016
00017  #include "gdcmTag.h"
00018
00019  #include <vector>
00020
00021  namespace gdcm

```

```

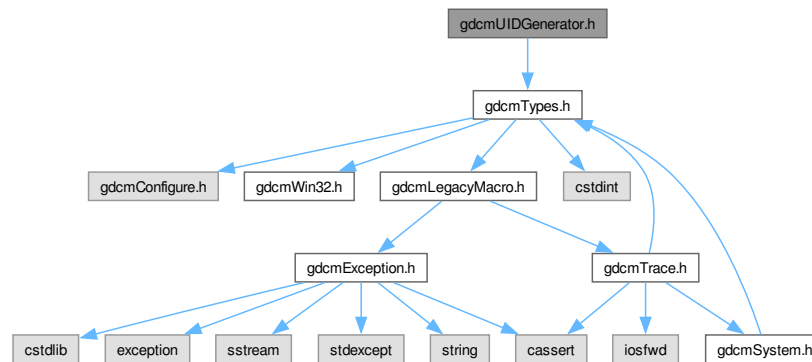
00022 {
00023
00030 class GDCM_EXPORT TagPath
00031 {
00032 public:
00033     TagPath();
00034     ~TagPath();
00035     void Print(std::ostream &) const;
00036
00041     bool ConstructFromString(const char *path);
00042
00044     static bool IsValid(const char *path);
00045
00047     bool ConstructFromTagList(Tag const *l, unsigned int n);
00048
00049     bool Push(Tag const & t);
00050     bool Push(unsigned int itemnum);
00051
00052 private:
00053     std::vector<Tag> Path;
00054 };
00055 } // end namespace gdc
00056 // end namespace gdc
00057
00058 #endif //GDCMTAGPATH_H

```

13.443 gdcUIDGenerator.h File Reference

#include "gdcTypes.h"

Include dependency graph for gdcUIDGenerator.h:



Classes

- class `gdc::UIDGenerator`
Class for generating unique UID.

Namespaces

- namespace `gdc`

13.444 gdcmlUIDGenerator.h

[Go to the documentation of this file.](#)

```

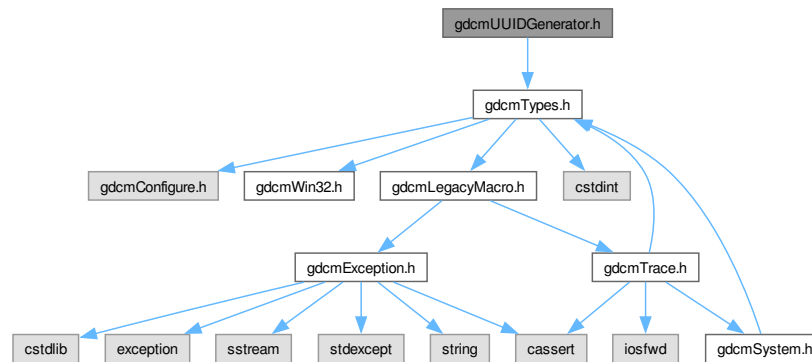
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMLUIDGENERATOR_H
00015 #define GDCMLUIDGENERATOR_H
00016
00017 #include "gdcmlTypes.h"
00018
00019 namespace gdcml
00020 {
00021
00022 class GDCML_EXPORT UIDGenerator
00023 {
00024 public:
00025     UIDGenerator():Unique() {}
00026
00027     // Function to override the GDCML root with a user one:
00028     // WARNING: This need to be a valid root, otherwise call will fail
00029     // Implementation note. According to DICOM standard PS 3.5, Section 9 :
00030     // Unique Identifiers (UIDs), we have:
00031     /*
00032     ...
00033     The <org root> portion of the UID uniquely identifies an organization, (i.e., manufacturer, research
00034     organization, NEMA, etc.), and is composed of a number of numeric components as defined by ISO 8824.
00035     The <suffix> portion of the UID is also composed of a number of numeric components, and shall be
00036     unique within the scope of the <org root>. This implies that the organization identified in the <org root> is
00037     responsible for guaranteeing <suffix> uniqueness by providing registration policies. These policies shall
00038     guarantee <suffix> uniqueness for all UID's created by that organization. Unlike the <org root>, which may
00039     be common for UID's in an organization, the <suffix> shall take different unique values between different
00040     UID's that identify different objects.
00041     ...
00042     */
00043     static void SetRoot(const char * root);
00044     static const char *GetRoot();
00045
00046     const char* Generate();
00047
00048     static bool IsValid(const char *uid);
00049
00050     static const char *GetGDCMLUID(); // who would want that in the public API ??
00051
00052 protected:
00053     static bool GenerateUUID(unsigned char *uuid_data);
00054
00055 private:
00056     static const char GDCML_UID[];
00057     static std::string Root;
00058     static std::string EncodedHardwareAddress;
00059     static std::string Unique; // Buffer
00060 };
00061
00062 } // end namespace gdcml
00063
00064 #endif //GDCMLUIDGENERATOR_H

```

13.445 gdcmUUIDGenerator.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class [gdcm::UUIDGenerator](#)
Class for generating unique UUID.

Namespaces

- namespace [gdcm](#)

13.446 gdcmUUIDGenerator.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMUUUIDGENERATOR_H
00015  #define GDCMUUUIDGENERATOR_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {

```

```

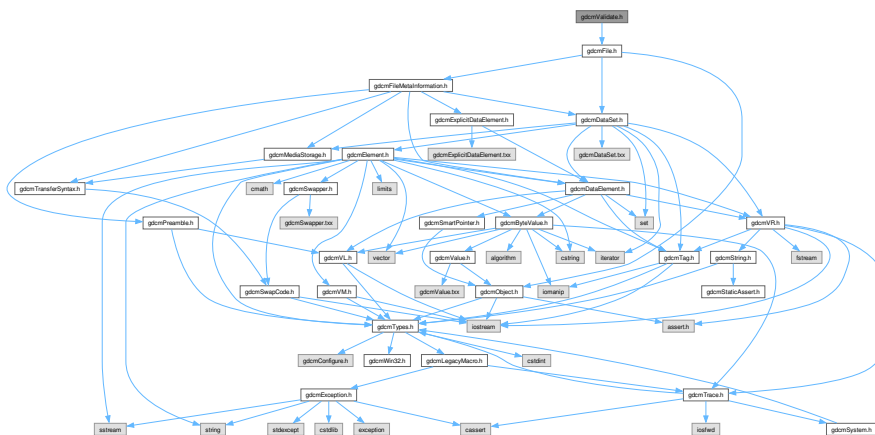
00021
00026 class GDCM_EXPORT UUIDGenerator
00027 {
00028 public:
00031     const char* Generate();
00032
00034     static bool IsValid(const char *uid);
00035
00036 private:
00037     std::string Unique; // Buffer
00038 };
00039
00040 } // end namespace gdcm
00041
00042 #endif //GDCMUUIDGENERATOR_H

```

13.447 gdcmValidate.h File Reference

#include "gdcmFile.h"

Include dependency graph for gdcmValidate.h:



Classes

- class `gdcm::Validate`
Validate class.

Namespaces

- namespace `gdcm`

13.448 gdcmlValidate.h

[Go to the documentation of this file.](#)

```

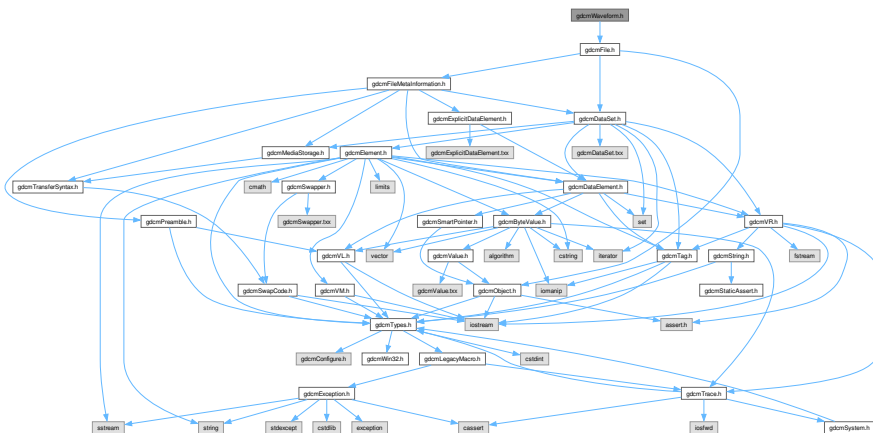
00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMVALIDATE_H
00015  #define GDCMVALIDATE_H
00016
00017  #include "gdcmFile.h"
00018
00019  namespace gdcm
00020  {
00021
00025  class GDCM_EXPORT Validate
00026  {
00027  public:
00028    Validate();
00029    ~Validate();
00030
00031    void SetFile(File const &f) { F = &f; }
00032    const File& GetValidatedFile() { return V; }
00033
00034    void Validation();
00035
00036  protected:
00037    const File *F;
00038    File V; // Validated file
00039  };
00040
00041  } // end namespace gdcm
00042
00043  #endif //GDCMVALIDATE_H

```

13.449 gdcMWaveform.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for `gdcMWaveform.h`:



Classes

- class [gdcm::Waveform](#)
 [Waveform](#) class.

Namespaces

- namespace [gdcm](#)

13.450 gdcmWaveform.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMWAVEFORM_H
00015  #define GDCMWAVEFORM_H
00016
00017  #include "gdcmFile.h"
00018
00019  namespace gdcm
00020  {
00021
00024  class GDCM_EXPORT Waveform
00025  {
00026  public:
00027    Waveform() = default;
00028
00029  private:
00030  };
00031
00032  } // end namespace gdcm
00033
00034  #endif //GDCMWAVEFORM_H

```

13.451 gdcmXMLPrinter.h File Reference

```

#include "gdcmFile.h"
#include "gdcmDataElement.h"

```



```

00026 # as an XML Element, since it does not appear in the binary encoded
00027 # DICOM objects. It exists here merely as a documentation aid.
00028
00029 DicomDataSet = DicomAttribute*
00030 DicomAttribute = element DicomAttribute {
00031   Tag, VR, Keyword?, PrivateCreator?,
00032   ( BulkData | Value+ | Item+ | PersonName+ )?
00033 }
00034
00035 BulkData = element BulkData{ UUID }
00036 Value = element Value { Number, xsd:string }
00037 Item = element Item { Number, DicomDataSet }
00038 PersonName = element PersonName {
00039   Number,
00040   element SingleByte { NameComponents }?,
00041   element Ideographic { NameComponents }?,
00042   element Phonetic
00043   { NameComponents }?
00044 }
00045
00046 NameComponents =
00047   element FamilyName {xsd:string}?,
00048   element GivenName {xsd:string}?,
00049   element MiddleName {xsd:string}?,
00050   element NamePrefix {xsd:string}?,
00051   element NameSuffix {xsd:string}?
00052
00053 # keyword is the attribute tag from PS3.6
00054 # (derived from the DICOM Attribute's name)
00055 Keyword = attribute keyword { xsd:token }
00056 # canonical XML definition of Hex, with lowercase letters disallowed
00057 Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
00058 VR = attribute vr { "AE" | "AS" | "AT" | "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
00059 | "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"
00060 | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
00061 PrivateCreator = attribute privateCreator{ xsd:string }
00062 UUID = attribute uuid { xsd:string }
00063 Number = attribute number { xsd:positiveInteger }
00064
00065
00066 */
00067
00068 #include "gdcmlFile.h"
00069 #include "gdcmlDataElement.h"
00070
00071 namespace gdcml
00072 {
00073
00074   class DataSet;
00075   class DictEntry;
00076   class Dicts;
00077
00078   class GDCM_EXPORT XMLPrinter
00079   {
00080   public:
00081     XMLPrinter();
00082     virtual ~XMLPrinter();
00083
00084     // Set file
00085     void SetFile(File const &f) { F = &f; }
00086
00087
00088
00089     typedef enum {
00090
00091         OnlyUUID = 0 ,
00092         LOADBULKDATA = 1
00093     } PrintStyles;
00094
00095
00096     // Set PrintStyle value
00097     void SetStyle(PrintStyles ps)
00098     {
00099         PrintStyle = ps;
00100     }
00101
00102     // Get PrintStyle value
00103     PrintStyles GetPrintStyle() const
00104     {
00105         return PrintStyle;
00106     }

```

```

00107
00108 // Print
00109 void Print(std::ostream& os);
00110
00111 // Print an individual dataset
00112 void PrintDataSet(const DataSet &ds, const TransferSyntax &ts, std::ostream& os);
00113
00114 //void PrintUID(std::ostream &os);
00115
00116 virtual void HandleBulkData(const char *uuid, const TransferSyntax &ts,
00117     const char *bulkdata, size_t bulklen);
00118
00119 protected:
00120
00121 VR PrintDataElement(std::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, const
00122     TransferSyntax &ts);
00123
00124 void PrintSQ(const SequenceOfItems *sqi, const TransferSyntax &ts, std::ostream &os);
00125
00126 PrintStyles PrintStyle;
00127
00128 const File *F;
00129
00130 };
00131
00132 } // end namespace gdcm
00133
00134 #endif //GDCMXMLPRINTER_H

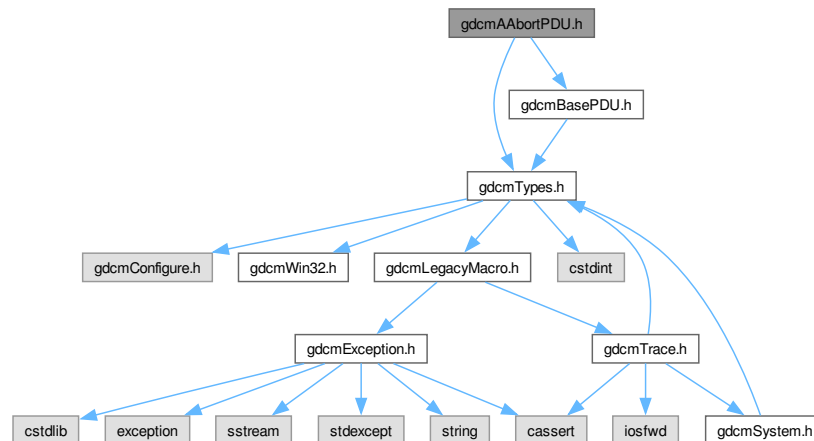
```

13.453 gdcmAAbortPDU.h File Reference

#include "gdcmTypes.h"

#include "gdcmBasePDU.h"

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class `gdcm::network::AAbortPDU`
`AAbortPDU`.

Namespaces

- namespace `gdcM`
- namespace `gdcM::network`

13.454 gdcmAAbortPDU.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMAABORTPDU_H
00015  #define GDCMAABORTPDU_H
00016
00017  #include "gdcMTypes.h"
00018  #include "gdcMBasePDU.h"
00019
00020  namespace gdcM
00021  {
00022
00023  namespace network
00024  {
00025
00030  class GDCM_EXPORT AAbortPDU : public BasePDU
00031  {
00032  public:
00033      AAbortPDU();
00034      std::istream &Read(std::istream &is) override;
00035      const std::ostream &Write(std::ostream &os) const override;
00036
00038      size_t Size() const override;
00039      void Print(std::ostream &os) const override;
00040
00041      bool IsLastFragment() const override { return true; }
00042
00043      void SetSource(const uint8_t s);
00044      void SetReason(const uint8_t r);
00045
00046  private:
00047      static const uint8_t ItemType; // PDUType ?
00048      static const uint8_t Reserved2;
00049      uint32_t ItemLength; // PDU Length
00050      static const uint8_t Reserved7;
00051      static const uint8_t Reserved8;
00052      uint8_t Source;
00053      uint8_t Reason; // diag
00054  };
00055
00056  } // end namespace network
00057
00058  } // end namespace gdcM
00059
00060  #endif //GDCMAABORTPDU_H

```

13.455 gdcMAAssociateACPDU.h File Reference

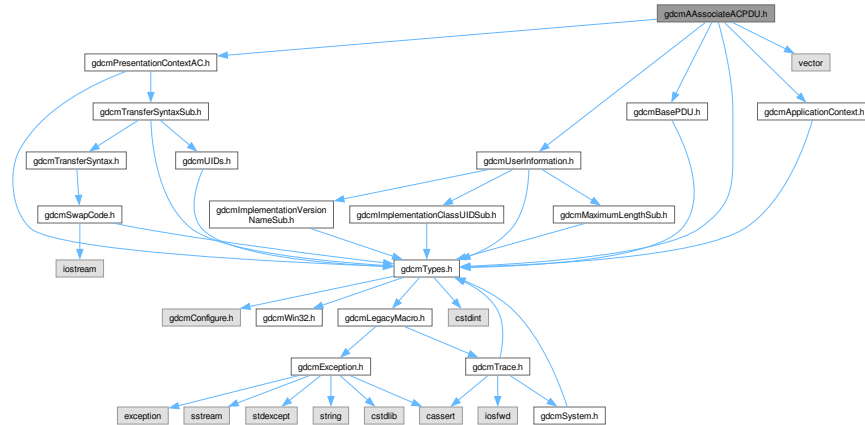
```

#include "gdcMTypes.h"
#include "gdcMApplicationContext.h"

```

```
#include "gdcmPresentationContextAC.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmAAssociateACPDU.h:



Classes

- class [gdcm::network::AAssociateACPDU](#)
[AAssociateACPDU](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.456 gdcmAAssociateACPDU.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMASSOCIATEACPDU_H
00015  #define GDCMASSOCIATEACPDU_H
00016
00017  #include "gdcmTypes.h"
```

```

00018 #include "gdcmApplicationContext.h"
00019 #include "gdcmPresentationContextAC.h"
00020 #include "gdcmUserInformation.h"
00021 #include "gdcmBasePDU.h"
00022
00023 #include <vector>
00024
00025 namespace gdcm
00026 {
00027
00028 namespace network
00029 {
00030 class AAssociateRQPDU;
00031
00037 class AAssociateACPDU : public BasePDU
00038 {
00039 public:
00040     AAssociateACPDU();
00041     std::istream &Read(std::istream &is) override;
00042     const std::ostream &Write(std::ostream &os) const override;
00043
00044     void AddPresentationContextAC( PresentationContextAC const &pcac );
00045
00046     typedef std::vector<PresentationContextAC>::size_type SizeType;
00047     const PresentationContextAC &GetPresentationContextAC( SizeType i ) {
00048         gdcm_assert( !PresContextAC.empty() && i < PresContextAC.size() );
00049         return PresContextAC[i];
00050     }
00051     SizeType GetNumberOfPresentationContextAC() const {
00052         return PresContextAC.size();
00053     }
00054     const UserInformation &GetUserInformation() const { return UserInfo; }
00055
00056     SizeType Size() const override;
00057
00058     void Print(std::ostream &os) const override;
00059     bool IsLastFragment() const override { return true; }
00060
00061     void InitFromRQ( AAssociateRQPDU const & rqpdu );
00062 protected:
00063     friend class AAssociateRQPDU;
00064     void SetCalledAETitle(const char calledaetitle[16]);
00065     void SetCallingAETitle(const char callingaetitle[16]);
00066
00067 private:
00068     void InitSimple( AAssociateRQPDU const & rqpdu );
00069
00070 private:
00071     static const uint8_t ItemType; // PDUType ?
00072     static const uint8_t Reserved2;
00073     uint32_t PDULength; // len of
00074     static const uint16_t ProtocolVersion;
00075     static const uint16_t Reserved9_10;
00076
00077     // This reserved field shall be sent with a value identical to the value
00078     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00079     // shall not be tested when received.
00080     char Reserved11_26[16];
00081     // This reserved field shall be sent with a value identical to the value
00082     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00083     // shall not be tested when received.
00084     char Reserved27_42[16];
00085     // This reserved field shall be sent with a value identical to the value
00086     // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
00087     // shall not be tested when received.
00088     char Reserved43_74[32];
00089     /*
00090     75-xxx Variable items This variable field shall contain the following items: one Application
00091     Context Item, one or more Presentation Context Item(s) and one User
00092     Information Item. For a complete description of these items see Sections
00093     7.1.1.2, 7.1.1.14, and 7.1.1.6.
00094     */
00095     ApplicationContext AppContext;
00096     std::vector<PresentationContextAC> PresContextAC;
00097     UserInformation UserInfo;
00098 };
00099
00100 } // end namespace network
00101
00102 } // end namespace gdcm
00103

```

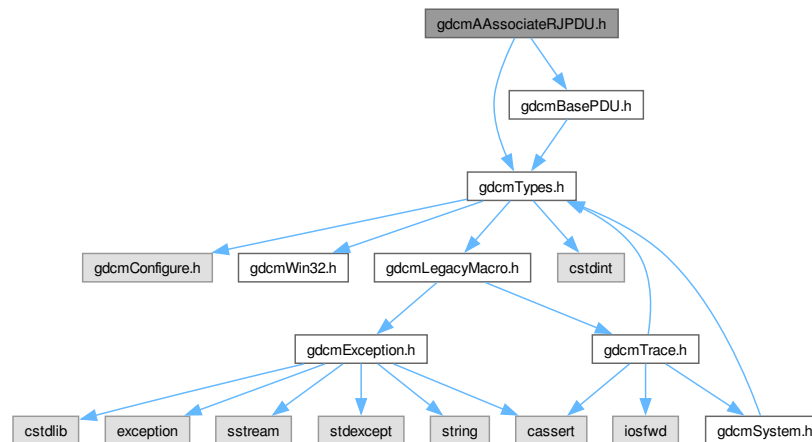
```
00104 #endif //GDCMASSOCIATEACPDU_H
```

13.457 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class [gdcm::network::AAssociateRJPDU](#)
[AAssociateRJPDU](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.458 gdcmAAssociateRJPDU.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
```



```

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE. See the above copyright notice for more information.
00012
00013      =====*/
00014 #ifndef GDCMAASSOCIATERJPDU_H
00015 #define GDCMAASSOCIATERJPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmBasePDU.h"
00019
00020 namespace gdcm
00021 {
00022
00023 namespace network
00024 {
00025
00031 class AAssociateRJPDU : public BasePDU
00032 {
00033 public:
00034     AAssociateRJPDU();
00035     std::istream &Read(std::istream &is) override;
00036     const std::ostream &Write(std::ostream &os) const override;
00037     void Print(std::ostream &os) const override;
00038     size_t Size() const override;
00039     bool IsLastFragment() const override { return true; }
00040 private:
00041     static const uint8_t ItemType; // PDUType ?
00042     static const uint8_t Reserved2;
00043     uint32_t ItemLength; // PDU Length ?
00044     static const uint8_t Reserved8;
00045     uint8_t Result;
00046     uint8_t Source;
00047     uint8_t Reason; // diag ?
00048 };
00049
00050 } // end namespace network
00051
00052 } // end namespace gdcm
00053
00054 #endif //GDCMAASSOCIATERJPDU_H

```

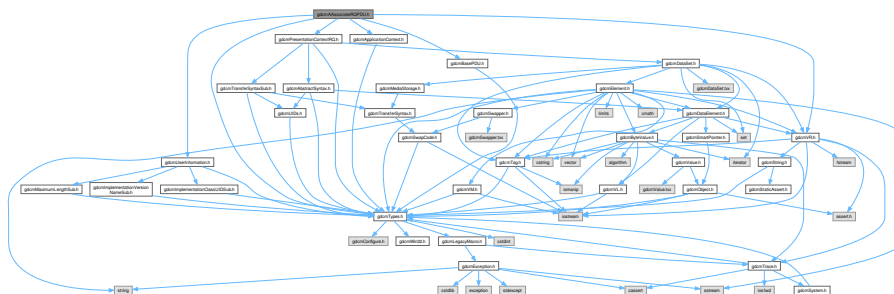
13.459 gdcmAAssociateRQPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"

```

Include dependency graph for gdcmAAssociateRQPDU.h:



Classes

- class `gdcm::network::AAssociateRQPDU`
`AAssociateRQPDU`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.460 `gdcmAAssociateRQPDU.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMAASSOCIATERQPDU_H
00015  #define GDCMAASSOCIATERQPDU_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmVR.h" // AEComp
00019  #include "gdcmApplicationContext.h"
00020  #include "gdcmPresentationContextRQ.h"
00021  #include "gdcmUserInformation.h"
00022  #include "gdcmBasePDU.h"
00023  00024  namespace gdcm
00025  {
00026  00027  namespace network
00028  {
00029  00030  class AAssociateACPDU;
00035  class AAssociateRQPDU : public BasePDU
00036  {
00037  public:
00038  AAssociateRQPDU();
00039  std::istream &Read(std::istream &is) override;
00040  const std::ostream &Write(std::ostream &os) const override;
00041  size_t Size() const override;
00042  void AddPresentationContext( PresentationContextRQ const &pc );
00043  00044  void SetCalledAETitle(const char calledaetitle[16]);
00046  std::string GetCalledAETitle() const { return std::string(CalledAETitle,16); }
00047  00048  void SetCallingAETitle(const char callingaetitle[16]);
00050  std::string GetCallingAETitle() const { return std::string(CallingAETitle,16); }
00051  00053  static bool IsAETitleValid(const char title[16]);
00054  00057  //void InitFromRQ( AAssociateACPDU &acpdu );
00058  00059  void Print(std::ostream &os) const override;
00060  00061  AAssociateRQPDU(const AAssociateRQPDU &pdu):BasePDU(pdu)
00062  {
00063  gdcm_assert( 0 );

```

```

00064     }
00065     //this function fails to compile on windows.
00066     // AAssociateRQPDU &operator=(const AAssociateRQPDU &_val)
00067     // {
00068     //     gdcmm_assert( 0 );
00069     // }
00070
00071     typedef std::vector<PresentationContextRQ>::size_type SizeType;
00072     SizeType GetNumberOfPresentationContext() const {
00073         return PresContext.size();
00074     }
00075     PresentationContextRQ const &GetPresentationContext(SizeType i) const {
00076         gdcmm_assert( !PresContext.empty() && i < PresContext.size() );
00077         return PresContext[i];
00078     }
00079     typedef std::vector<PresentationContextRQ> PresentationContextArrayType;
00080     PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }
00081
00082     const PresentationContextRQ *GetPresentationContextByID(uint8_t i) const;
00083     const PresentationContextRQ *GetPresentationContextByAbstractSyntax(AbstractSyntax const & absyn ) const;
00084     bool IsLastFragment() const override { return true; }
00085
00086     const UserInformation & GetUserInformation() const { return UserInfo; }
00087     void SetUserInformation( UserInformation const & ui );
00088
00089 protected:
00090     friend class AAssociateACPDU;
00091     std::string GetReserved43_74() const;
00092
00093 private:
00094     // 1 PDU-type 01H
00095     static const uint8_t ItemType; // PDUType ?
00096     // 2 Reserved This reserved field shall be sent with a value 00H but not tested to this value when received.
00097     static const uint8_t Reserved2;
00098     /* 3-6 PDU-length This PDU-length shall be the number of bytes from the first byte of the
00099        following field to the last byte of the variable field. It shall be encoded as
00100        an unsigned binary number
00101        */
00102     uint32_t ItemLength; // PDU Length
00103     /*
00104     7-8 Protocol-version This two byte field shall use one bit to identify each version of the
00105     DICOM UL protocol supported by the calling end-system. This is
00106     Version 1 and shall be identified with bit 0 set. A receiver of this PDU
00107     implementing only this version of the DICOM UL protocol shall only test
00108     that bit 0 is set.
00109     */
00110     static const uint16_t ProtocolVersion;
00111     /*
00112     9-10 Reserved This reserved field shall be sent with a value 0000H but not tested to
00113     this value when received.
00114     */
00115     static const uint16_t Reserved9_10;
00116     /*
00117     11-26 Called-AE-title Destination DICOM Application Name. It shall be encoded as 16
00118     characters as defined by the ISO 646:1990-Basic G0 Set with leading
00119     and trailing spaces (20H) being non-significant. The value made of 16
00120     spaces (20H) meaning "no Application Name specified" shall not be
00121     used. For a complete description of the use of this field, see Section
00122     7.1.1.4.
00123     */
00124     char CalledAETitle[16];
00125     /*
00126     27-42 Calling-AE-title Source DICOM Application Name. It shall be encoded as 16
00127     characters as defined by the ISO 646:1990-Basic G0 Set with leading
00128     and trailing spaces (20H) being non-significant. The value made of 16
00129     spaces (20H) meaning "no Application Name specified" shall not be
00130     used. For a complete description of the use of this field, see Section
00131     7.1.1.3.
00132     */
00133     char CallingAETitle[16];
00134     /*
00135     43-74 Reserved This reserved field shall be sent with a value 00H for all bytes but not
00136     tested to this value when received
00137     */
00138     char Reserved43_74[32]; // { 0 }
00139     /*
00140     75-xxx Variable items This variable field shall contain the following items: one Application
00141     Context Item, one or more Presentation Context Items and one User
00142     Information Item. For a complete description of the use of these items
00143     see Sections 7.1.1.2, 7.1.1.13, and 7.1.1.6.
00144     */

```


Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.462 gdcmAbstractSyntax.h

[Go to the documentation of this file.](#)

```

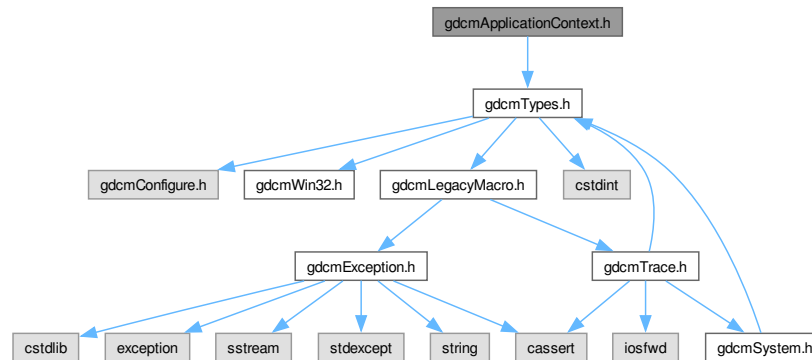
00001
00002  /*=====
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMABSTRACTSYNTAX_H
00015 #define GDCMABSTRACTSYNTAX_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmUIDs.h"
00019 #include "gdcmDataElement.h"
00020
00021 namespace gdcm
00022 {
00023
00024     namespace network
00025     {
00026
00032         class AbstractSyntax
00033         {
00034         public:
00035             AbstractSyntax();
00036             std::istream &Read(std::istream &is);
00037             const std::ostream &Write(std::ostream &os) const;
00038
00039             void SetName( const char *name ) { UpdateName( name ); }
00040             const char *GetName() const { return Name.c_str(); }
00041
00042             // accept a UID's::TSType also...
00043             void SetNameFromUID( UIDs::TSName tname );
00044             //now that the PresentationContext messes around with UIDs and returns a string
00045             //use that string as well.
00046             //void SetNameFromUIDString( const std::string& inUIDName );
00047
00048             size_t Size() const;
00049
00050             void Print(std::ostream &os) const;
00051
00052             bool operator==(const AbstractSyntax & as) const
00053             {
00054                 return Name == as.Name;
00055             }
00056
00057             DataElement GetAsDataElement() const;
00058
00059         private:
00060             void UpdateName( const char *name );
00061             static const uint8_t ItemType;
00062             static const uint8_t Reserved2;
00063             uint16_t ItemLength; // len of
00064             std::string /*AbstractSyntax*/ Name; // UID
00065         };
00066
00067     } // end namespace network
00068 } // end namespace gdcm
00069
00070 #endif //GDCMABSTRACTSYNTAX_H

```

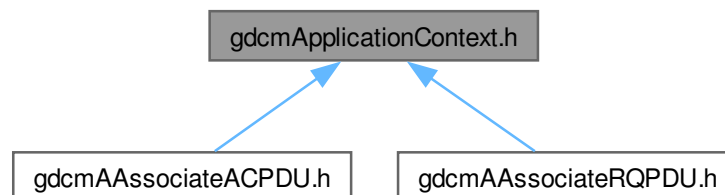
13.463 gdcmApplicationContext.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmApplicationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ApplicationContext](#)
[ApplicationContext](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.464 gdcmApplicationContext.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMAPPLICATIONCONTEXT_H
00015 #define GDCMAPPLICATIONCONTEXT_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00032 class ApplicationContext
00033 {
00034 public:
00035     ApplicationContext();
00036     std::istream &Read(std::istream &is);
00037     const std::ostream &Write(std::ostream &os) const;
00038
00039     void SetName( const char *name ) { UpdateName( name ); }
00040     const char *GetName() const { return Name.c_str(); }
00041     size_t Size() const;
00042
00043     //static const uint8_t GetItemType() { return ItemType; }
00044     void Print(std::ostream &os) const;
00045
00046 private:
00047     void UpdateName( const char *name );
00048     static const uint8_t ItemType;
00049     static const uint8_t Reserved2;
00050     uint16_t ItemLength; // len of application context name
00051     std::string /*ApplicationContext*/ Name; // UID
00052 };
00053
00054 } // end namespace network
00055
00056 } // end namespace gdcm
00057
00058 #endif //GDCMAPPLICATIONCONTEXT_H

```

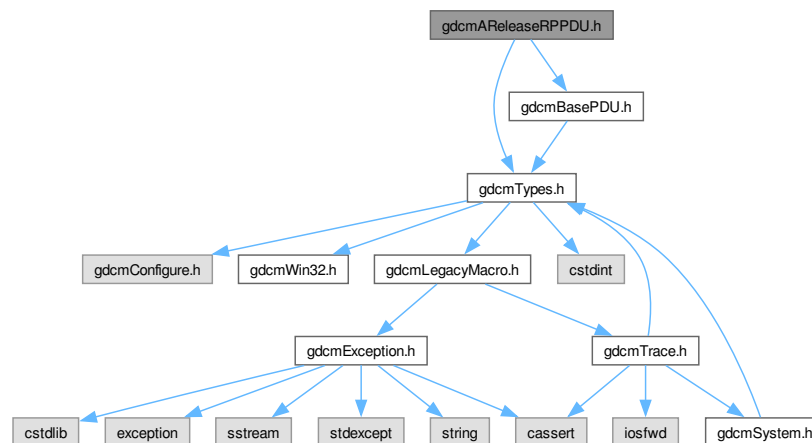
13.465 gdcmAReleaseRPPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

```

Include dependency graph for `gdcmAReleaseRPPDU.h`:



Classes

- class `gdcm::network::AReleaseRPPDU`
`AReleaseRPPDU`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.466 `gdcmAReleaseRPPDU.h`

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMARELEASERPPDU_H
00015  #define GDCMARELEASERPPDU_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmBasePDU.h"
00019
00020  namespace gdcm

```



```

00021 {
00022
00023 namespace network
00024 {
00025
00031 class AReleaseRPPDU : public BasePDU
00032 {
00033 public:
00034     AReleaseRPPDU();
00035     std::istream &Read(std::istream &is) override;
00036     const std::ostream &Write(std::ostream &os) const override;
00037     size_t Size() const override;
00038     void Print(std::ostream &os) const override;
00039     bool IsLastFragment() const override { return true; }
00040 private:
00041     static const uint8_t ItemType; // PDUType ?
00042     static const uint8_t Reserved2;
00043     uint32_t ItemLength; // PDU Length
00044     static const uint32_t Reserved7_10;
00045 };
00046
00047 } // end namespace network
00048
00049 } // end namespace gdcm
00050
00051 #endif //GDCMARELEASERPPDU_H

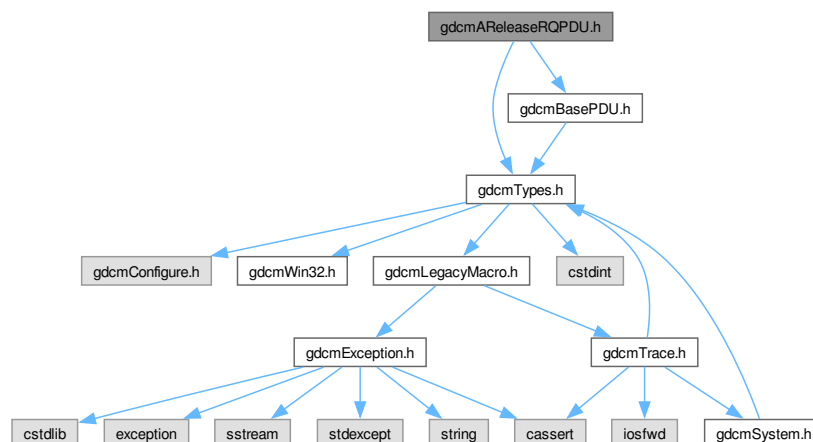
```

13.467 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRQPDU.h:



Classes

- class `gdcm::network::AReleaseRPPDU`
`AReleaseRPPDU`.

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.468 gdcmAReleaseRQPDU.h

[Go to the documentation of this file.](#)

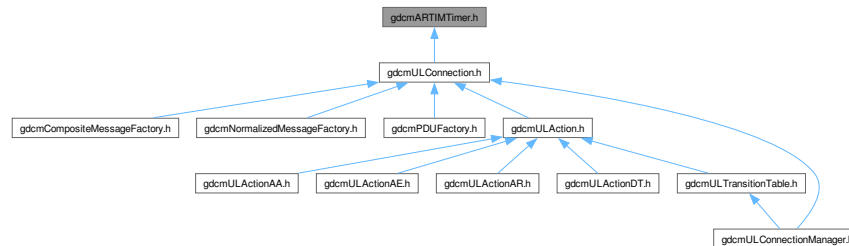
```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMARELEASERQPDU_H
00015 #define GDCMARELEASERQPDU_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmBasePDU.h"
00019
00020 namespace gdcm
00021 {
00022
00023 namespace network
00024 {
00025
00031 class AReleaseRQPDU : public BasePDU
00032 {
00033 public:
00034   AReleaseRQPDU();
00035   std::istream &Read(std::istream &is) override;
00036   const std::ostream &Write(std::ostream &os) const override;
00037   size_t Size() const override;
00038   void Print(std::ostream &os) const override;
00039   bool IsLastFragment() const override { return true; }
00040 private:
00041   static const uint8_t ItemType; // PDUType ?
00042   static const uint8_t Reserved2;
00043   uint32_t ItemLength; // PDU Length
00044   static const uint32_t Reserved7_10;
00045 };
00046
00047 } // end namespace network
00048
00049 } // end namespace gdcm
00050
00051 #endif //GDCMARELEASERQPDU_H

```

13.469 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ARTIMTimer](#)
[ARTIMTimer](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.470 gdcmARTIMTimer.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 * Copyright NumFOCUS
00004 *
00005 * Licensed under the Apache License, Version 2.0 (the "License");
00006 * you may not use this file except in compliance with the License.
00007 * You may obtain a copy of the License at
00008 *
00009 *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010 *
00011 * Unless required by applicable law or agreed to in writing, software
00012 * distributed under the License is distributed on an "AS IS" BASIS,
00013 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014 * See the License for the specific language governing permissions and
00015 * limitations under the License.
00016 *
00017 *=====*/
00018 #ifndef GDCMARTIMTIMER_H
00019 #define GDCMARTIMTIMER_H
00020
00021 namespace gdcm {
00022     namespace network{
00023         class ARTIMTimer
00024         {
00025         private:
00026             double mStartTime; //ms timing should be good enough, but there are also
00027             //high-resolution timing options. Those return doubles. For now,

```

```

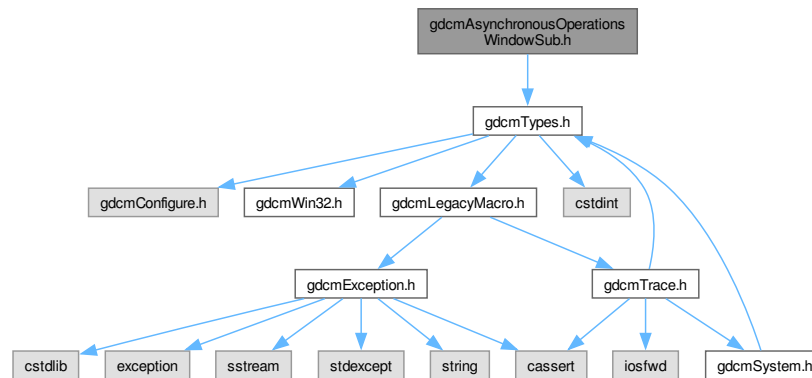
00043 //go with integer timing solutions based on milliseconds (DWORD on windows),
00044 //but leave as doubles to ease transitions to other timing methods.
00045
00046 double mTimeOut;
00047 //once GetCurrentTime() -mStartTime > mTimeout, GetHasExpired returns true.
00048
00049 double GetCurrentTime() const;//a platform-specific implementation of getting the
00050 //current time.
00051
00052 public:
00053 ARTIMTimer(); //initiates the start and timeout at -1;
00054 void Start(); //start' the timer by getting the current wall time
00055 void Stop(); //stop' the timer by resetting the 'start' to -1;
00056 void SetTimeout(double inTimeout);
00057 double GetTimeout() const;
00058
00059 double GetElapsedTime() const;
00060
00061 bool GetHasExpired() const;
00062
00063 };
00064 }
00065 }
00066
00067 #endif //GDCMARTIMTIMER_H

```

13.471 gdcmAsynchronousOperationsWindowSub.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class `gdcm::network::AsynchronousOperationsWindowSub`
`AsynchronousOperationsWindowSub`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.472 gdcmAsynchronousOperationsWindowSub.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
00015  #define GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
00016  00017  #include "gdcmTypes.h"
00018  00019  namespace gdcm
00020  {
00021  00022  namespace network
00023  {
00024  00032  class AsynchronousOperationsWindowSub
00033  {
00034  public:
00035  AsynchronousOperationsWindowSub();
00036  std::istream &Read(std::istream &is);
00037  const std::ostream &Write(std::ostream &os) const;
00038  00039  size_t Size() const;
00040  void Print(std::ostream &os) const;
00041  00042  private:
00043  static const uint8_t ItemType;
00044  static const uint8_t Reserved2;
00045  uint16_t ItemLength;
00046  uint16_t MaximumNumberOperationsInvoked;
00047  uint16_t MaximumNumberOperationsPerformed;
00048  };
00049  00050  } // end namespace network
00051  00052  } // end namespace gdcm
00053  00054  #endif // GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H

```

13.473 gdcmBaseCompositeMessage.h File Reference

```

#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>

```

[illegible]

```

graph BT
    gdcBaseCompositeMessage.h --> gdcmCEchoMessages.h
    gdcBaseCompositeMessage.h --> gdcmCFindMessages.h
    gdcBaseCompositeMessage.h --> gdcmCMoveMessages.h
    gdcBaseCompositeMessage.h --> gdcmCStoreMessages.h
  
```

- class `gdcn::network::BaseCompositeMessage`
BaseCompositeMessage.

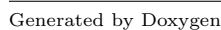
- namespace `gdcm`
- namespace `gdcm::network`

[Go to the documentation of this file.](#)

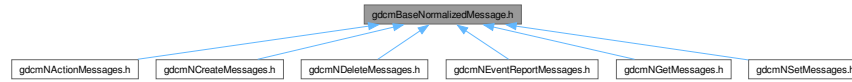
Generated by Doxygen

13.475 [gdcmBaseNormalizedMessage.h File Reference](#)

Include dependency graph for `gdcmbaseNormalizedMessage.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmb::network::BaseNormalizedMessage`
`BaseNormalizedMessage`.

Namespaces

- namespace `gdcmb`
- namespace `gdcmb::network`

13.476 gdcmbBaseNormalizedMessage.h

[Go to the documentation of this file.](#)

```

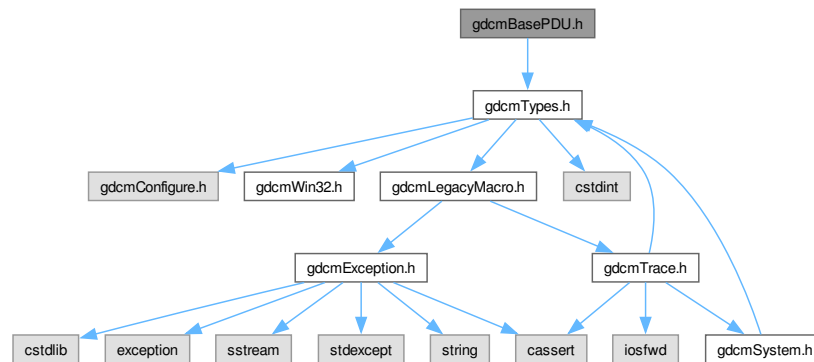
00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcmb.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMBASENORMALIZEDMESSAGE_H
00015  #define GDCMBASENORMALIZEDMESSAGE_H
00016
00017  #include "gdcmbPresentationDataValue.h"
00018  #include "gdcmbBaseQuery.h"
00019
00020  #include <vector>
00021
00022  namespace gdcmb
00023  {
00024      namespace network
00025      {
00026          class ULConnection;
00049          class BaseNormalizedMessage
00050          {
00051          public:
00052              virtual ~BaseNormalizedMessage() = default;
00053              //construct the appropriate pdv and dataset for this message
00054              //for instance, setting tag 0x0,0x100 to the appropriate value
00055              //the pdv, as described in Annex E of 3.8-2009, is the first byte
00056              //of the message (the MessageHeader), and then the subsequent dataset
00057              //that describes the operation.
00058              virtual std::vector<PresentationDataValue> ConstructPDV( const ULConnection &inConnection,
00059                  const BaseQuery * inQuery) = 0;
00060          };
00061      }
00062  }
00063  #endif //GDCMBASENORMALIZEDMESSAGE_H

```

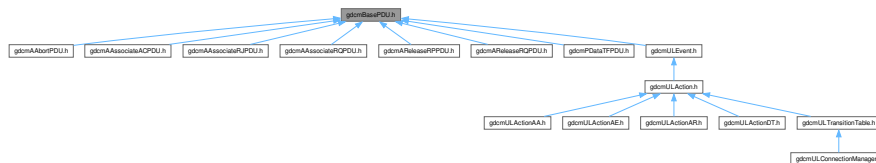

13.477 gdcmBasePDU.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBasePDU.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::network::BasePDU`
`BasePDU`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.478 gdcmBasePDU.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003  *
00004  * Copyright NumFOCUS
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMBASEPDU_H
00019 #define GDCMBASEPDU_H
00020
00021 #include "gdcmTypes.h"
00022
00023 namespace gdcm
00024 {
00025     namespace network
00026     {
00027
00050         class BasePDU
00051         {
00052         public:
00053             virtual ~BasePDU() = default;
00054
00055             virtual std::istream &Read(std::istream &is) = 0;
00056             virtual const std::ostream &Write(std::ostream &os) const = 0;
00057
00058             virtual size_t Size() const = 0;
00059             virtual void Print(std::ostream &os) const = 0;
00060
00061             virtual bool IsLastFragment() const = 0;
00062         };
00063
00064     } // end namespace network
00065 } // end namespace gdcm
00066
00067 #endif // GDCMBASEPDU_H

```

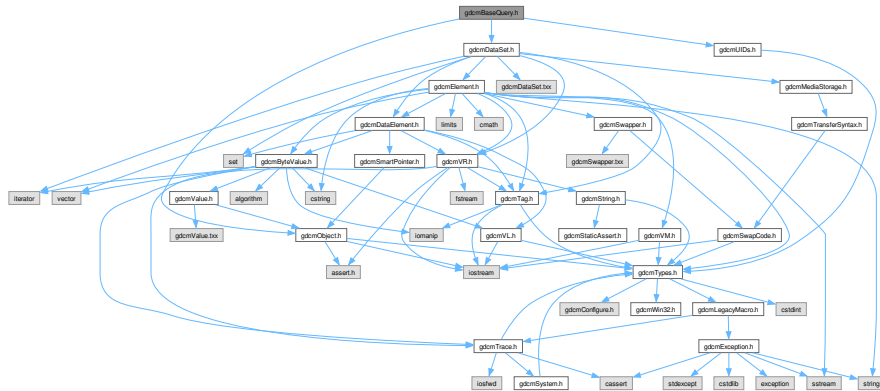
13.479 gdcmBaseQuery.h File Reference

```

#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"

```

Include dependency graph for gdcmbaseQuery.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdc::BaseQuery`
BaseQuery.

Namespaces

- namespace `gdcm`

Enumerations

- enum `gdc::ENQueryType` {
`gdc::eCreateMMPS` = 0 ,
`gdc::eSetMMPS` }

13.480 gdcmBaseQuery.h

[Go to the documentation of this file.](#)

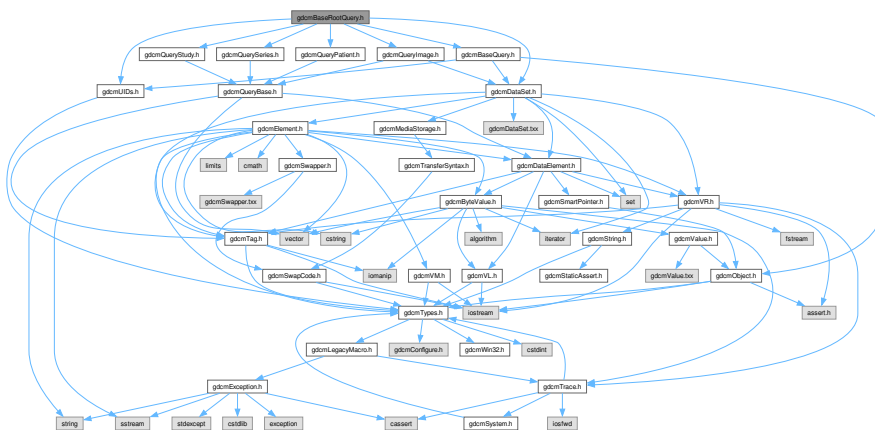
```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMBASEQUERY_H
00019 #define GDCMBASEQUERY_H
00020
00021 #include "gdcmDataSet.h"
00022 #include "gdcmUIDs.h"
00023 #include "gdcmObject.h"
00024
00025 namespace gdcm
00026 {
00027     class QueryFactory;
00028     class DictEntry;
00029
00030     enum ENQueryType
00031     {
00032         eCreateMMPS = 0,
00033         eSetMMPS
00034     };
00041     class GDCM_EXPORT BaseQuery : public Object
00042     {
00043     public:
00044         //these four classes contain the required, unique, and optional tags from the standard.
00045         //used both to list the tags as well as to validate a dataset, if ever we were to do so.
00046         protected:
00047             DataSet mDataSet;
00048             friend class QueryFactory;
00049             BaseQuery();
00050
00051             std::string mSopInstanceUID;
00052
00053             void SetSearchParameter(const Tag& inTag, const DictEntry& inDictEntry, const std::string& inValue);
00054
00055             bool ValidDataSet( const DataSet & dataSetToValid, const DataSet & dataSetReference ) const ;
00056         public:
00057             ~BaseQuery() override;
00058
00059             void SetSearchParameter(const Tag& inTag, const std::string& inValue);
00060             void SetSearchParameter(const std::string& inKeyword, const std::string& inValue);
00061
00062             const std::ostream &WriteHelpFile(std::ostream &os);
00063
00064             //this function allows writing of the query to disk for storing for future use
00065             //virtual in case it needs to be overridden
00066             //returns false if the operation failed
00067             bool WriteQuery(const std::string& inFileName);
00068
00069             DataSet const & GetQueryDataSet() const;
00070             DataSet & GetQueryDataSet();
00071             void AddQueryDataSet(const DataSet & ds);
00072
00073             virtual bool ValidateQuery( bool inStrict = true ) const = 0;
00074
00075             virtual UIDs::TSName GetAbstractSyntaxUID() const = 0;
00076             std::string GetSOPInstanceUID() const { return mSopInstanceUID ; }
00077             void SetSOPInstanceUID( const std::string & iSopInstanceUID ) { mSopInstanceUID = iSopInstanceUID ; }
00078
00079
00080

```

13.481 gdcmbaseRootQuery.h File Reference

Include dependency graph for gdcmBaseRootQuery.h:



```

graph TD
    Root["glibmmBaseProcQuery.h"]
    Root --> B["glibmmBaseProcMessage.h"]
    Root --> C["glibmmCompressNetworkFunctions.h"]
    Root --> D["glibmmFindParentProcQuery.h"]
    Root --> E["glibmmFindStubProcQuery.h"]
    Root --> F["glibmmMoveStubProcQuery.h"]
    Root --> G["glibmmQueryFactory.h"]
    Root --> H["glibmmWMFindQuery.h"]
    
    B --> I["glibmmCFindMessages.h"]
    B --> J["glibmmCStubMessages.h"]
    B --> K["glibmmCStubMessages.h"]
    B --> L["glibmmMoveStubMessages.h"]
    
    D --> M["glibmmMoveStubProcQuery.h"]
    E --> M
    F --> M
    
```

- class `gdc::BaseRootQuery`
`BaseRootQuery`.

- namespace `gdcm`

Enumerations

- enum `gdcm::EQueryLevel` {
`gdcm::ePatient` = 0 ,
`gdcm::eStudy` = 1 ,
`gdcm::eSeries` = 2 ,
`gdcm::eImage` = 3 }
- enum `gdcm::EQueryType` {
`gdcm::eFind` = 0 ,
`gdcm::eMove` ,
`gdcm::eWLMFind` }

13.482 `gdcmBaseRootQuery.h`

[Go to the documentation of this file.](#)

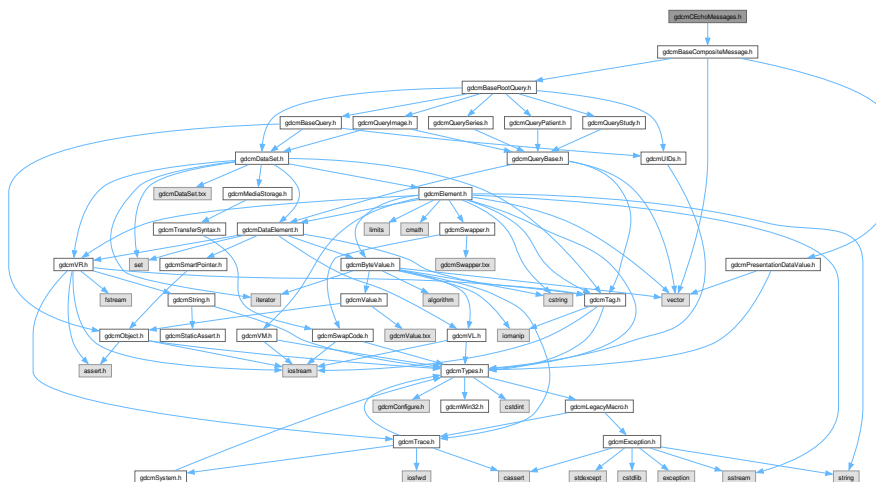
```

00001
00002  /*=====
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMBASEROOTQUERY_H
00019  #define GDCMBASEROOTQUERY_H
00020
00021  #include "gdcmDataSet.h"
00022  #include "gdcmUIDs.h"
00023  #include "gdcmBaseQuery.h"
00024  #include "gdcmQueryPatient.h"
00025  #include "gdcmQueryStudy.h"
00026  #include "gdcmQuerySeries.h"
00027  #include "gdcmQueryImage.h"
00028
00029  namespace gdcm
00030  {
00031      class QueryFactory;
00032      class DictEntry;
00033
00034      enum EQueryLevel
00035      {
00036          // -1 is reserved do not use
00037          ePatient = 0,
00038          eStudy = 1,
00039          eSeries = 2,
00040          eImage = 3
00041      };
00042      enum EQueryType
00043      {
00044          eFind = 0,
00045          eMove,
00046          eWLMFind
00047      };
00048
00066      class GDCM_EXPORT BaseRootQuery : public BaseQuery
00067      {
00068          //these four classes contain the required, unique, and optional tags from the standard.
00069          //used both to list the tags as well as to validate a dataset, if ever we were to do so.
00070      protected:

```

13.483 gdcMCEchoMessages.h File Reference

Include dependency graph for gdcMCEchoMessages.h:



- class `gdcm::network::CEchoRQ`
`CEchoRQ`.
- class `gdcm::network::CEchoRSP`
`CEchoRSP` this file defines the messages for the cecho action.

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.484 gdcmCEchoMessages.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCECHOMESSAGES_H
00019 #define GDCMCECHOMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022
00023 namespace gdcm{
00024     namespace network{
00025
00026     class ULConnection;
00027
00032     class CEchoRQ : public BaseCompositeMessage {
00033     public:
00034         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00035             const BaseRootQuery* inRootQuery) override;
00036     };
00037
00042     class CEchoRSP : public BaseCompositeMessage {
00043     public:
00044         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00045     };
00046     }
00047 }
00048 #endif // GDCMCECHOMESSAGES_H

```

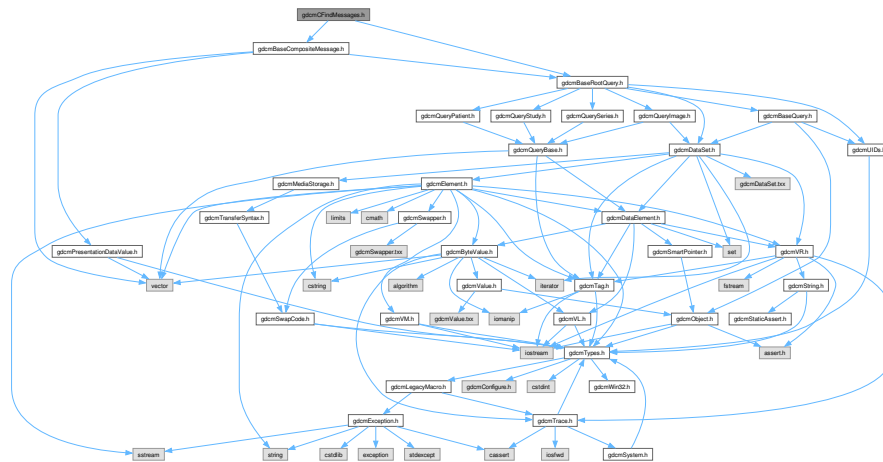
13.485 gdcmCFindMessages.h File Reference

```

#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"

```


Include dependency graph for gdcmCFindMessages.h:



Classes

- class [gdcm::network::CFindCancelRQ](#)
[CFindCancelRQ](#) this file defines the messages for the cfind action.
- class [gdcm::network::CFindRQ](#)
[CFindRQ](#).
- class [gdcm::network::CFindRSP](#)
[CFindRSP](#) this file defines the messages for the cfind action.

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.486 gdcmCFindMessages.h

[Go to the documentation of this file.](#)

```
00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  */
```

13.487 [gdcmCMoveMessages.h](#) File Reference

- class `gdcmm::network::CMoveCancelRq`
- class `gdcmm::network::CMoveRQ`

[CMoveRQ](#).

- class [gdcm::network::CMoveRSP](#)

[CMoveRSP](#) this file defines the messages for the cmove action.

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.488 gdcmCMoveMessages.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCMOVEMESSAGES_H
00019 #define GDCMCMOVEMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022 #include "gdcmBaseRootQuery.h"
00023
00024 namespace gdcm{
00025     namespace network{
00026         class ULConnection;
00027         class CMoveRQ : public BaseCompositeMessage {
00028             //this class will fulfill the inheritance,
00029             //but additional information is needed by cmovd
00030             //namely, the root type or the calling AE-TITLE
00031             std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00032         public:
00033             std::vector<PresentationDataValue> ConstructPDV(
00034                 const ULConnection &inConnection,
00035                 const BaseRootQuery* inRootQuery) override;
00036         };
00037         class CMoveRSP : public BaseCompositeMessage {
00038         public:
00039             std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00040         };
00041         class CMoveCancelRq : public BaseCompositeMessage {
00042         public:
00043             std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00044         };
00045     }
00046 }
00047 #endif

```



```

00010     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011     PURPOSE. See the above copyright notice for more information.
00012
00013
00014     =====*/
00014 #ifndef GDCMCOMMANDDATASET_H
00015 #define GDCMCOMMANDDATASET_H
00016
00017 #include "gdcmDataSet.h"
00018 #include "gdcmDataElement.h"
00019
00020 namespace gdcm
00021 {
00022 class GDCM_EXPORT CommandDataSet : public DataSet
00023 {
00024 public:
00025     CommandDataSet() = default;
00026     ~CommandDataSet() = default;
00027
00028     friend std::ostream &operator<<(std::ostream &_os, const CommandDataSet &_val);
00029
00030     // FIXME: no virtual function means: duplicate code...
00031     void Insert(const DataElement& de) {
00032         if( de.GetTag().GetGroup() == 0x0000 )
00033         {
00034             InsertDataElement( de );
00035         }
00036         else
00037         {
00038             gdcmErrorMacro( "Cannot add element with group != 0x0000 in the command dataset : " << de );
00039         }
00040     }
00041     void Replace(const DataElement& de) {
00042         Remove(de.GetTag());
00043         Insert(de);
00044     }
00045
00046     std::istream &Read(std::istream &is);
00047
00048     std::ostream &Write(std::ostream &os) const;
00049
00050 protected:
00051 };
00052 //-----
00053 inline std::ostream& operator<<(std::ostream &os, const CommandDataSet &val)
00054 {
00055     val.Print( os );
00056     return os;
00057 }
00058
00059 } // end namespace gdcm
00060
00061 #endif //GDCMFILEMETAINFORMATION_H

```

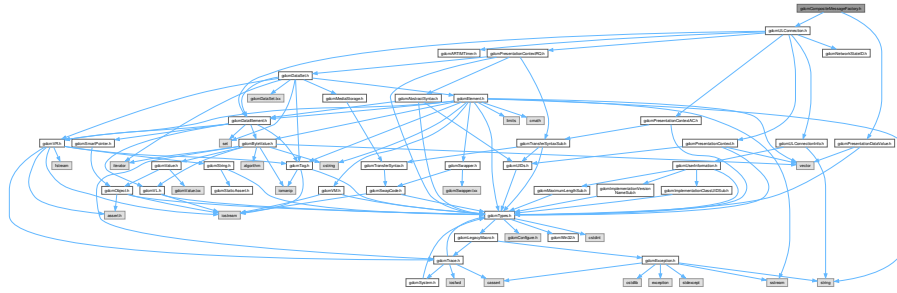
13.491 gdcmCompositeMessageFactory.h File Reference

```

#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"

```

Include dependency graph for `gdcCompositeMessageFactory.h`:



Classes

- class `gdc::network::CompositeMessageFactory`
`CompositeMessageFactory`.

Namespaces

- namespace `gdc`
- namespace `gdc::network`

13.492 `gdcCompositeMessageFactory.h`

[Go to the documentation of this file.](#)

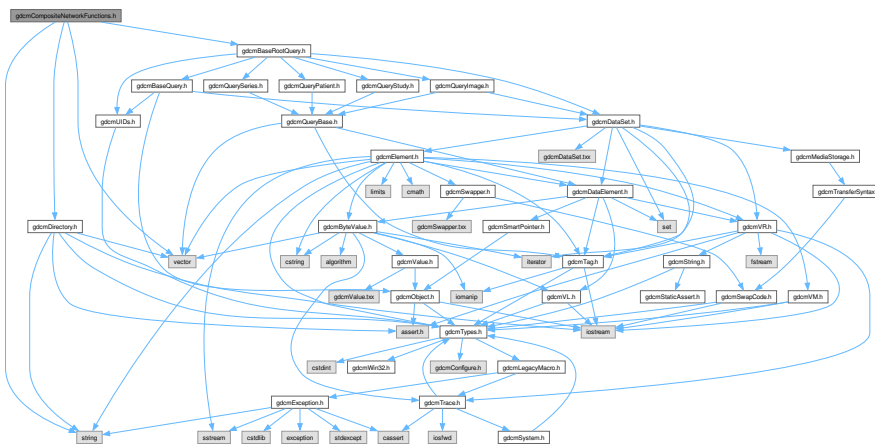
```

00001  /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMCOMPOSITEMESSAGEFACTORY_H
00019  #define GDCMCOMPOSITEMESSAGEFACTORY_H
00020
00021  #include "gdcPresentationDataValue.h"
00022  #include "gdcULConnection.h"
00023
00024  namespace gdc {
00025  class BaseRootQuery;
00026  class File;
00027  namespace network {
00028  class BasePDU;
00037  class CompositeMessageFactory
00038  {
00039  public:

```

13.493 [gdcmCompositeNetworkFunctions.h](#) File Reference

Include dependency graph for gdcmCompositeNetworkFunctions.h:



- class `gdc::CompositeNetworkFunctions`
Composite Network Functions.

- namespace `gdcm`

13.494 gdcmCompositeNetworkFunctions.h

[Go to the documentation of this file.](#)

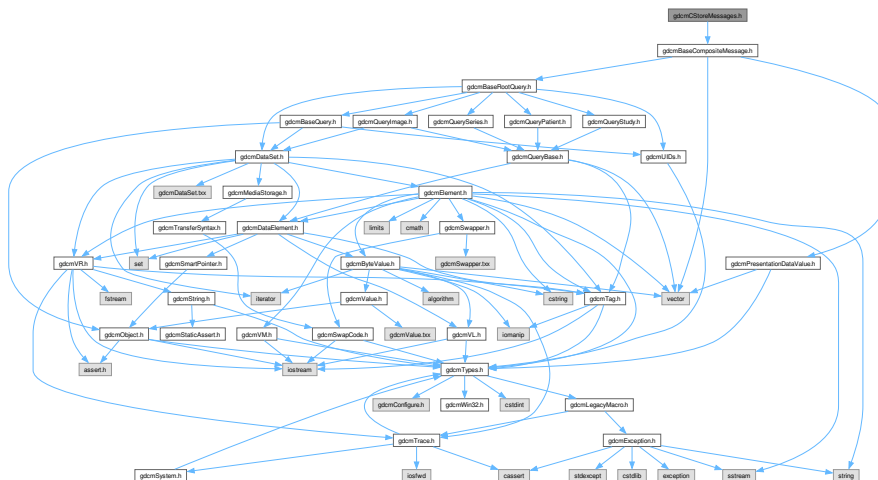
```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMCOMPOSITENETWORKFUNCTIONS_H
00019 #define GDCMCOMPOSITENETWORKFUNCTIONS_H
00020
00021 #include "gdcmDirectory.h"
00022 #include "gdcmBaseRootQuery.h" // EQueryLevel / EQueryType
00023
00024 #include <vector>
00025 #include <string>
00026
00027 namespace gdcm
00028 {
00029     class GDCM_EXPORT CompositeNetworkFunctions
00030     {
00031     public:
00032         static bool CEcho( const char *remote, uint16_t portno, const char *aetitle = nullptr,
00033             const char *call = nullptr );
00034
00035         typedef std::pair<Tag, std::string> KeyValuePairType;
00036         typedef std::vector< KeyValuePairType > KeyValuePairArrayType;
00037
00038         static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
00039             const DataSet& queryds, EQueryType queryType = eFind );
00040
00041         static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
00042             const KeyValuePairArrayType& keys, EQueryType queryType = eFind );
00043
00044         static bool CMove( const char *remote, uint16_t portno, const BaseRootQuery* query,
00045             uint16_t portsep, const char *aetitle = nullptr,
00046             const char *call = nullptr, const char *outputdir = nullptr);
00047
00048         static bool CFind( const char *remote, uint16_t portno,
00049             const BaseRootQuery* query,
00050             std::vector<DataSet> &retDataSets,
00051             const char *aetitle = nullptr,
00052             const char *call = nullptr );
00053
00054         static bool CStore( const char *remote, uint16_t portno,
00055             const Directory::FileNamesType & filenames,
00056             const char *aetitle = nullptr, const char *call = nullptr);
00057     };
00058 }
00059 // end namespace gdcm
00060 #endif // GDCMCOMPOSITENETWORKFUNCTIONS_H

```



```
#include "gdcmBaseCompositeMessage.h"
Include dependency graph for gdcmCStoreMessages.h:
```



- class `gdcmm::network::CStoreRQ`
`CStoreRQ`.
- class `gdcmm::network::CStoreRSP`
`CStoreRSP` this file defines the messages for the cecho action.

- namespace `gdcm`
- namespace `gdcm::network`

[Go to the documentation of this file.](#)

```
00001 /*=====
00002 *
00003 * Copyright NumFOCUS
00004 *
00005 * Licensed under the Apache License, Version 2.0 (the "License");
00006 * you may not use this file except in compliance with the License.
00007 * You may obtain a copy of the License at
00008 *
00009 * http://www.apache.org/licenses/LICENSE-2.0.txt
00010 *
00011 * Unless required by applicable law or agreed to in writing, software
00012 * distributed under the License is distributed on an "AS IS" BASIS,
00013 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```

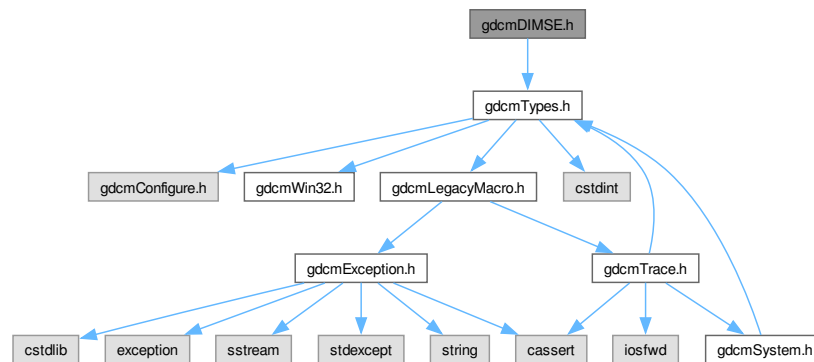
00014 * See the License for the specific language governing permissions and
00015 * limitations under the License.
00016 *
00017 *=====*/
00018 #ifndef GDCMCSTOREMESSAGES_H
00019 #define GDCMCSTOREMESSAGES_H
00020
00021 #include "gdcmBaseCompositeMessage.h"
00022
00023 namespace gdcm{
00024 class File;
00025 namespace network{
00026 class BasePDU;
00031 class CStoreRQ : public BaseCompositeMessage {
00032 std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery*
inRootQuery) override;//to fulfill the virtual contract
00033 public:
00034 std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00035 const File& file, bool writeDataSet = true );
00036 };
00037
00042 class CStoreRSP : public BaseCompositeMessage {
00043 std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery*
inRootQuery) override;//to fulfill the virtual contract
00044 public:
00045 std::vector<PresentationDataValue> ConstructPDV(const DataSet* inDataSet, const BasePDU* inPC);
00046 };
00047 }
00048 }
00049 #endif // GDCMCSTOREMESSAGES_H

```

13.497 gdcmDIMSE.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmDIMSE.h:



Classes

- class [gdcm::network::CEchoRQ](#)
[CEchoRQ](#).
- class [gdcm::network::CEchoRSP](#)
[CEchoRSP](#) this file defines the messages for the cecho action.
- class [gdcm::network::CFind](#)
- class [gdcm::network::DIMSE](#)
[DIMSE](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.498 gdcmDIMSE.h

[Go to the documentation of this file.](#)

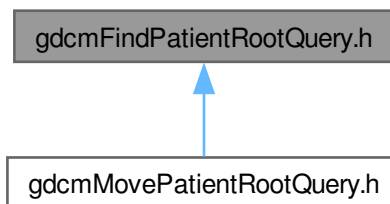
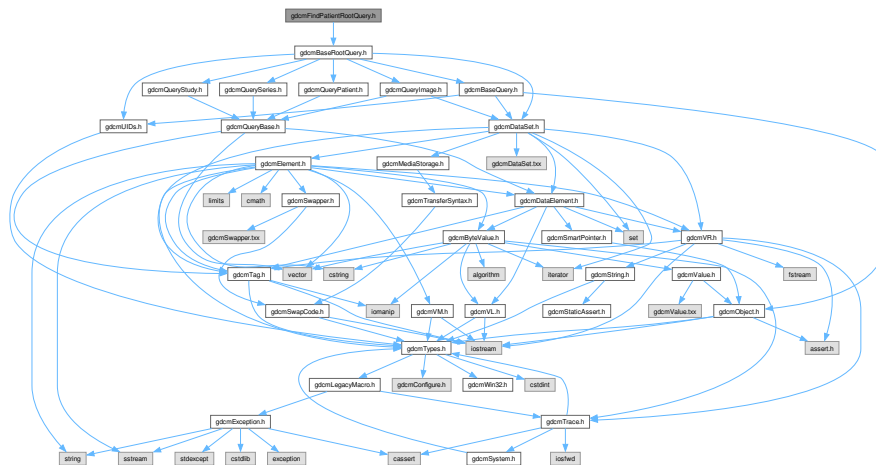
```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMDIMSE_H
00015 #define GDCMDIMSE_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00025 class DIMSE {
00026 public:
00027     typedef enum {
00028         C_STORE_RQ          = 0x0001,
00029         C_STORE_RSP         = 0x8001,
00030         C_GET_RQ            = 0x0010,
00031         C_GET_RSP           = 0x8010,
00032         C_FIND_RQ           = 0x0020,
00033         C_FIND_RSP          = 0x8020,
00034         C_MOVE_RQ           = 0x0021,
00035         C_MOVE_RSP          = 0x8021,
00036         C_ECHO_RQ           = 0x0030,
00037         C_ECHO_RSP          = 0x8030,
00038         N_EVENT_REPORT_RQ   = 0x0100,
00039         N_EVENT_REPORT_RSP  = 0x8100,
00040         N_GET_RQ            = 0x0110,
00041         N_GET_RSP           = 0x8110,
00042         N_SET_RQ            = 0x0120,
00043         N_SET_RSP           = 0x8120,
00044         N_ACTION_RQ         = 0x0130,
00045         N_ACTION_RSP        = 0x8130,
00046         N_CREATE_RQ         = 0x0140,
00047         N_CREATE_RSP        = 0x8140,
00048         N_DELETE_RQ         = 0x0150,
00049         N_DELETE_RSP        = 0x8150,
00050         C_CANCEL_RQ         = 0x0FFF
00051     } CommandTypes;
00052 };
00053
00054 /*
00055 9.1.5.1 C-ECHO parameters
00056 Table 9.1-5
00057 C-ECHO PARAMETERS
00058 */
00059 class CEchoRQ
00060 {
00061 public:
00062     uint16_t      MessageID;
00063     UIComp        AffectedSOPClassUID;
00064
00065     /* M */
00066     /* M */

```

```
00072 };
00073
00074 class CEchoRSP
00075 {
00076 public:
00077 /*
00078 Message ID M U
00079 Message ID Being Responded To M
00080 Affected SOP Class UID M U(=)
00081 Status M
00082 */
00083 };
00084
00085 class CFind
00086 {
00087 /*
00088 Failure Refused: Out of Resources A700 (0000,0902)
00089 Identifier does not match SOP Class A900 (0000,0901)
00090 (0000,0902)
00091 Unable to process Cxxx (0000,0901)
00092 (0000,0902)
00093 Cancel Matching terminated due to Cancel
00094 request
00095 FE00 None
00096 Success Matching is complete – No final Identifier
00097 is supplied.
00098 0000 None
00099 Pending Matches are continuing – Current Match
00100 is supplied and any Optional Keys were
00101 supported in the same manner as
00102 Required Keys.
00103 FF00 Identifier
00104 Matches are continuing – Warning that
00105 one or more Optional Keys were not
00106 supported for existence and/or matching
00107 for this Identifier.
00108 FF01 Identifier
00109 */
00110 };
00111
00112
00113 } // end namespace network
00114
00115 } // end namespace gdcm
00116
00117 #endif //GDCMDIMSE_H
```

```
#include "gdcmBaseRootQuery.h"
```



Classes

- class `gdcmm::FindPatientRootQuery`
PatientRootQuery.

Namespaces

- namespace `gdcm`

[Go to the documentation of this file.](#)

13.501 [gdcMFindStudyRootQuery.h](#) File Reference

Classes

- class [gdcm::FindStudyRootQuery](#)
[FindStudyRootQuery](#).

Namespaces

- namespace [gdcm](#)

13.502 gdcmFindStudyRootQuery.h

[Go to the documentation of this file.](#)

```

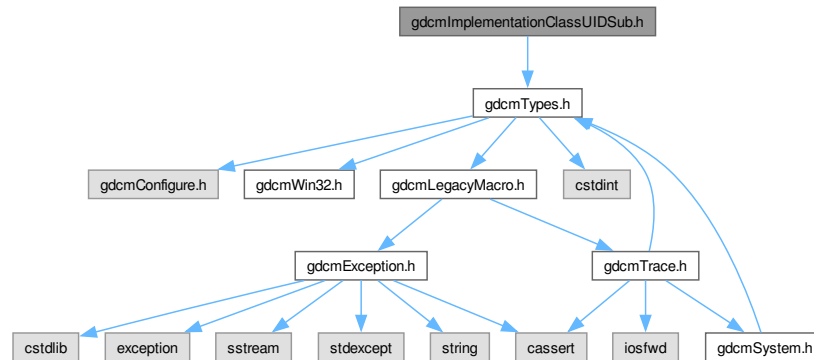
00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013  #ifndef GDCMFINDSTUDYROOTQUERY_H
00014  #define GDCMFINDSTUDYROOTQUERY_H
00015
00016  #include "gdcmBaseRootQuery.h"
00017
00018  namespace gdcm
00019  {
00020  class GDCM_EXPORT FindStudyRootQuery : public BaseRootQuery
00021  {
00022  friend class QueryFactory;
00023  public:
00024  FindStudyRootQuery();
00025
00026  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00027
00028  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00029
00030  bool ValidateQuery(bool inStrict = true) const override;
00031
00032  UIDs::TSName GetAbstractSyntaxUID() const override;
00033  };
00034
00035  } // end namespace gdcm
00036
00037  #endif // GDCMFINDSTUDYROOTQUERY_H

```

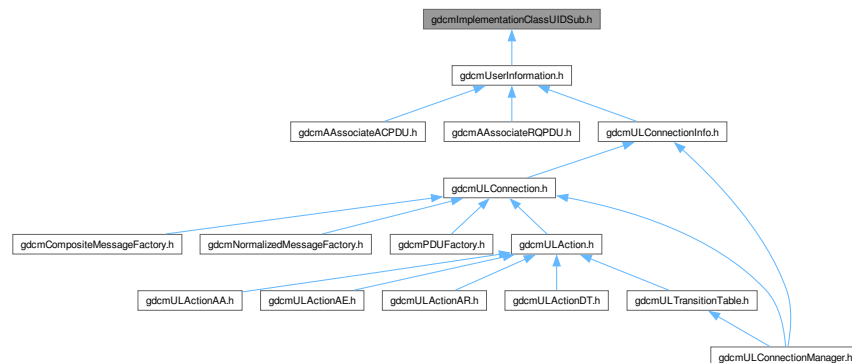
13.503 gdcmImplementationClassUIDSub.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmImplementationClassUIDSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::network::ImplementationClassUIDSub`
`ImplementationClassUIDSub`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.504 gdcmImplementationClassUIDSub.h

[Go to the documentation of this file.](#)

```

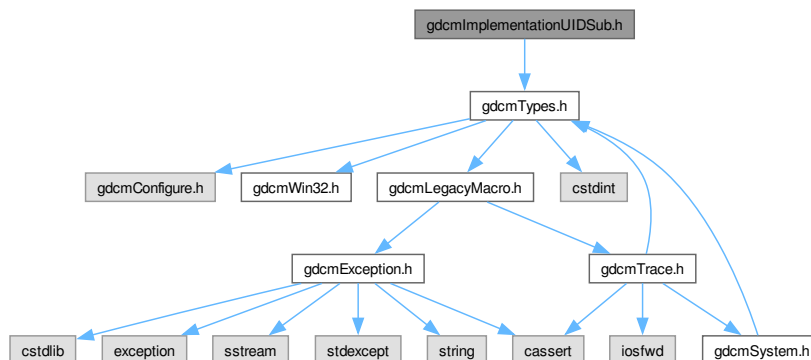
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMPLEMENTATIONCLASSUIDSUB_H
00015 #define GDCMIMPLEMENTATIONCLASSUIDSUB_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00025 class ImplementationClassUIDSub
00026 {
00027 public:
00028     ImplementationClassUIDSub();
00029     std::istream &Read(std::istream &is);
00030     const std::ostream &Write(std::ostream &os) const;
00031
00032     size_t Size() const;
00033
00034     void Print(std::ostream &os) const;
00035
00036 private:
00037     static const uint8_t ItemType;
00038     static const uint8_t Reserved2;
00039     uint16_t ItemLength;
00040     std::string ImplementationClassUID;
00041 };
00042
00043 } // end namespace network
00044
00045 } // end namespace gdcm
00046
00047 #endif //GDCMMAXIMUMLENGTHSUB_H

```

13.505 gdcmImplementationUIDSub.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class [gdcm::network::ImplementationUIDSub](#)
[ImplementationUIDSub](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.506 gdcmImplementationUIDSub.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMPLEMENTATIONUIDSUB_H
00015 #define GDCMIMPLEMENTATIONUIDSUB_H
00016
00017 #include "gdcmTypes.h"
00018

```

```

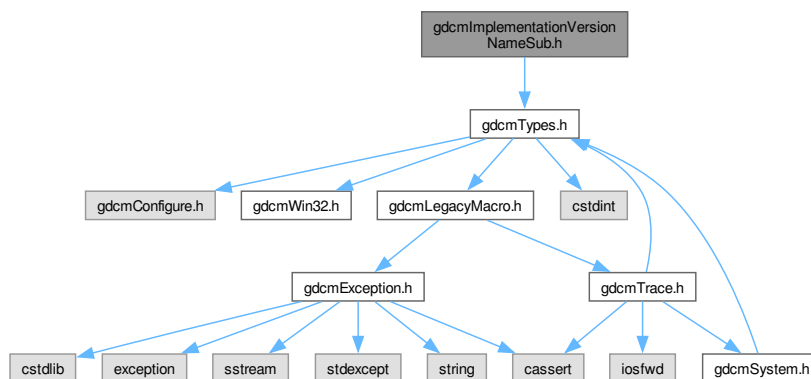
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00030 class GDCM__EXPORT ImplementationUIDSub
00031 {
00032 public:
00033     ImplementationUIDSub();
00034     const std::ostream &Write(std::ostream &os) const;
00035 private:
00036     static const uint8_t ItemType;
00037     static const uint8_t Reserved2;
00038     uint16_t ItemLength;
00039     std::string ImplementationClassUID;
00040 };
00041
00042 } // end namespace network
00043
00044 } // end namespace gdcm
00045
00046 #endif //GDCMMAXIMUMLENGTHSUB_H

```

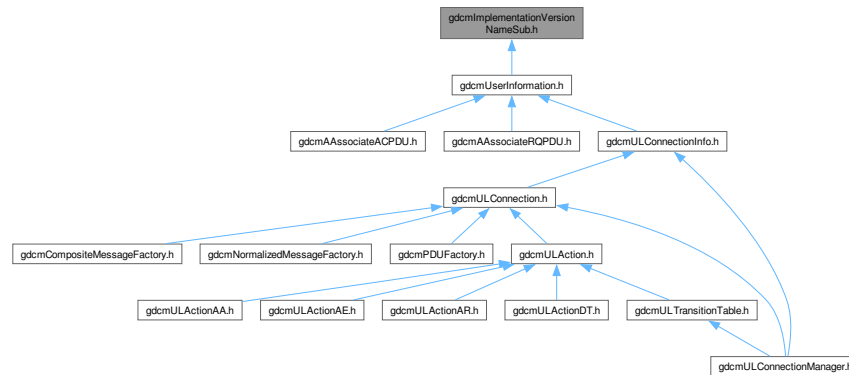
13.507 gdcmImplementationVersionNameSub.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmImplementationVersionNameSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcml::network::ImplementationVersionNameSub`
`ImplementationVersionNameSub`.

Namespaces

- namespace `gdcml`
- namespace `gdcml::network`

13.508 gdcmlImplementationVersionNameSub.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMIMPLEMENTATIONVERSIONNAMESUB_H
00015 #define GDCMIMPLEMENTATIONVERSIONNAMESUB_H
00016
00017 #include "gdcmlTypes.h"
00018
00019 namespace gdcml
00020 {
00021
00022 namespace network
00023 {
00024
00030 class ImplementationVersionNameSub

```

```

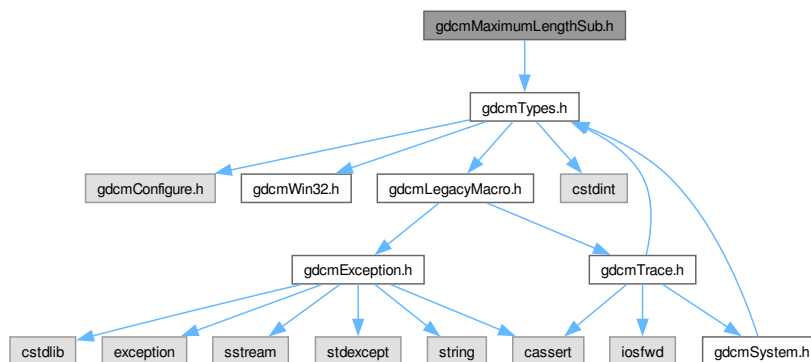
00031 {
00032 public:
00033     ImplementationVersionNameSub();
00034     std::istream &Read(std::istream &is);
00035     const std::ostream &Write(std::ostream &os) const;
00036
00037     size_t Size() const;
00038     void Print(std::ostream &os) const;
00039
00040 private:
00041     static const uint8_t ItemType;
00042     static const uint8_t Reserved2;
00043     uint16_t ItemLength;
00044     std::string ImplementationVersionName;
00045 };
00046
00047 } // end namespace network
00048
00049 } // end namespace gdcm
00050
00051 #endif //GDCMMAXIMUMLNGTHSUB_H

```

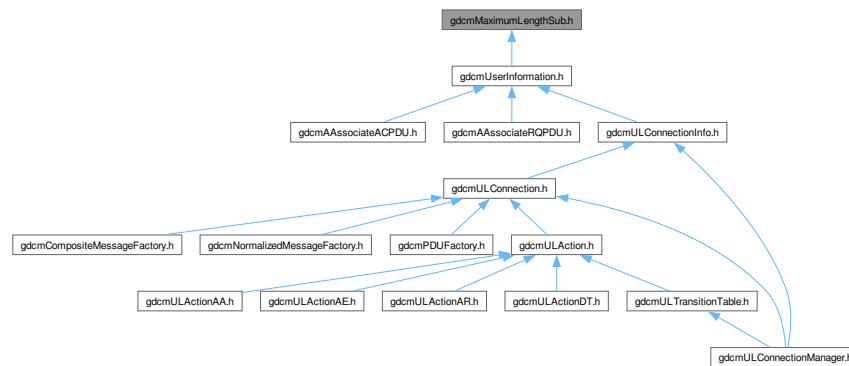
13.509 gdcmMaximumLengthSub.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmMaximumLengthSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::MaximumLengthSub](#)
[MaximumLengthSub](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.510 gdcmMaximumLengthSub.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMMAXIMUMLENGTHSUB_H
00015  #define GDCMMAXIMUMLENGTHSUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021
00022  namespace network
00023  {
00024
00036  class MaximumLengthSub

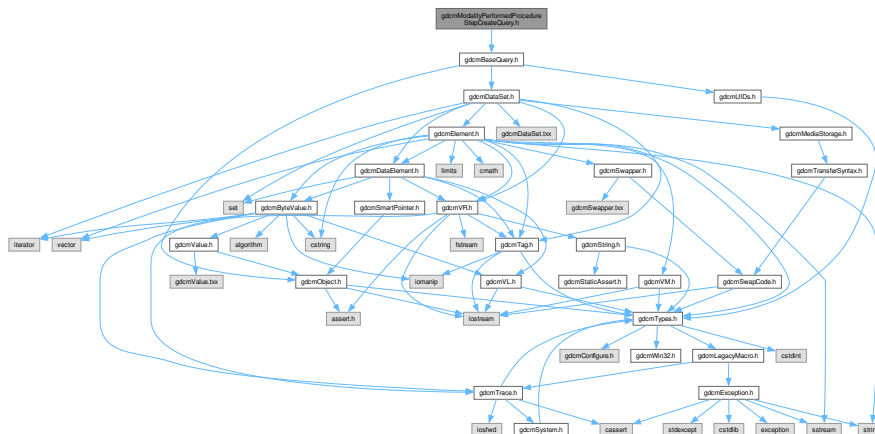
```

```
00037 {
00038     public:
00039         MaximumLengthSub();
00040         std::istream &Read(std::istream &is);
00041         const std::ostream &Write(std::ostream &os) const;
00042
00043         size_t Size() const;
00044
00045         uint32_t GetMaximumLength() const { return MaximumLength; }
00046         void SetMaximumLength(uint32_t maximumlength);
00047
00048         void Print(std::ostream &os) const;
00049
00050     private:
00051         static const uint8_t ItemType;
00052         static const uint8_t Reserved2;
00053         uint16_t ItemLength;
00054         uint32_t MaximumLength;
00055 };
00056
00057 } // end namespace network
00058
00059 } // end namespace gdcmm
00060
00061 #endif //GDCMMAXIMUMLENGTHSUB H
```

13.511 [gdcmModalityPerformedProcedureStepCreateQuery.h](#) File Reference

```
#include "gdcmBaseQuery.h"
```

Include dependency graph for gdcModalityPerformedProcedureStepCreateQuery.h:



Classes

- class `gdcmm::ModalityPerformedProcedureStepCreateQuery`
`ModalityPerformedProcedureStepCreateQuery`.

Namespaces

- namespace `gdcm`

Classes

- class [gdcm::ModalityPerformedProcedureStepSetQuery](#)
[ModalityPerformedProcedureStepSetQuery](#).

Namespaces

- namespace [gdcm](#)

13.514 gdcmModalityPerformedProcedureStepSetQuery.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013  #ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
00014  #define GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
00015
00016  #include "gdcmBaseQuery.h"
00017
00018  namespace gdcm
00019  {
00020  {
00021  class GDCM_EXPORT ModalityPerformedProcedureStepSetQuery : public BaseQuery{
00022  friend class QueryFactory;
00023  public:
00024  ModalityPerformedProcedureStepSetQuery( const std::string & iSopInstanceUID );
00025
00026  gdcm::DataSet GetRequiredDataSet() const;
00027  bool ValidateQuery(bool inStrict = true) const override;
00028  UIDs::TSName GetAbstractSyntaxUID() const override;
00029  };
00030  } // end namespace gdcm
00031
00032  #endif // GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H

```



```

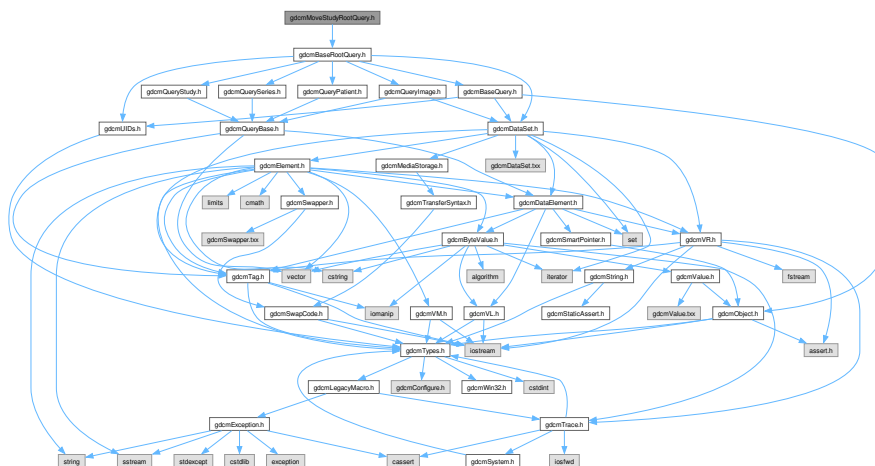
00017 #include "gdcFindPatientRootQuery.h"
00018
00019 namespace gdc
00020 {
00025 class GDCM_EXPORT MovePatientRootQuery : public BaseRootQuery
00026 {
00027     friend class QueryFactory;
00028 public:
00029     MovePatientRootQuery();
00030
00031     void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00032
00033     std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00034
00035     bool ValidateQuery(bool inStrict = true) const override;
00036
00037     UIDs::TSName GetAbstractSyntaxUID() const override;
00038 };
00039
00040 } // end namespace gdc
00041
00042 #endif // GDCMMOVEPATIENTROOTQUERY_H

```

13.517 gdcmoveStudyRootQuery.h File Reference

#include "gdcBaseRootQuery.h"

Include dependency graph for gdcmoveStudyRootQuery.h:



Classes

- class [gdc::MoveStudyRootQuery](#)
MoveStudyRootQuery.

Namespaces

- namespace [gdc](#)

Classes

- class `gdcm::network::NActionRQ`
`NActionRQ`.
- class `gdcm::network::NActionRSP`
`NActionRSP` this file defines the messages for the NAction action.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.520 gdcmNActionMessages.h

[Go to the documentation of this file.](#)

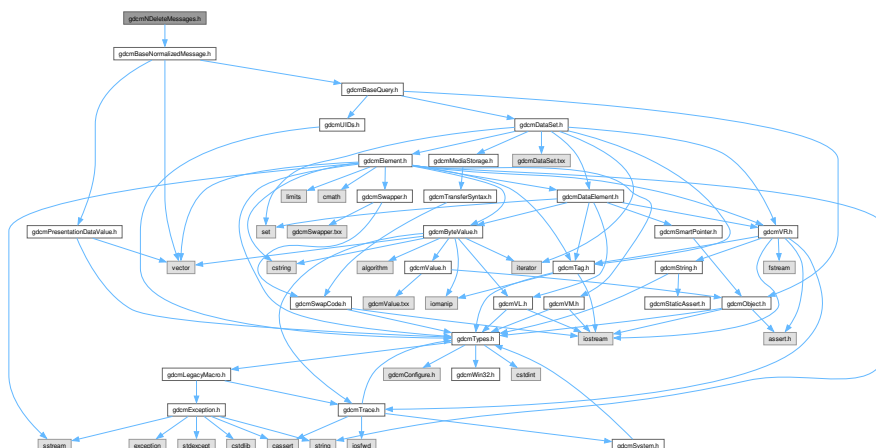
```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMCNACTIONMESSAGES_H
00015  #define GDCMCNACTIONMESSAGES_H
00016
00017  #include "gdcmBaseNormalizedMessage.h"
00018
00019  namespace gdcm{
00020  namespace network{
00021
00022  class ULConnection;
00023
00028  class NActionRQ : public BaseNormalizedMessage {
00029  public:
00030      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031      const BaseQuery* inQuery) override;
00032  };
00033
00038  class NActionRSP : public BaseNormalizedMessage {
00039  public:
00040      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041  };
00042  }
00043  }
00044  #endif // GDCMCNACTIONMESSAGES_H

```


13.523 gdcMDeleteMessages.h File Reference

Include dependency graph for gdcMDeleteMessages.h:



NDeleteRSP this file defines the messages for the ndelete action.

- namespace `gdcm`
- namespace `gdcm::network`

13.524 gdcmNDeleteMessages.h

[Go to the documentation of this file.](#)

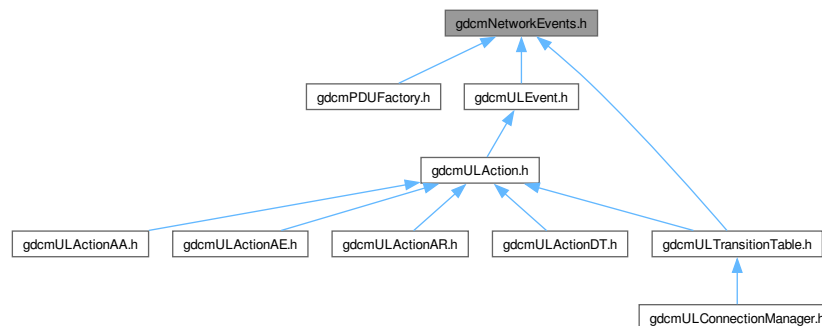
```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCNDELETEMESSAGES_H
00015 #define GDCMCNDELETEMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020     namespace network{
00021
00022         class ULConnection;
00023
00024         class NDeleteRQ : public BaseNormalizedMessage {
00025         public:
00026             std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00027                 const BaseQuery* inQuery) override;
00028         };
00029
00030         class NDeleteRSP : public BaseNormalizedMessage {
00031         public:
00032             std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00033         };
00034     }
00035 }
00036 #endif // GDCMCNDELETEMESSAGES_H

```

13.525 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

Enumerations

- enum gdcm::network::EEventID {
gdcm::network::eAASSOCIATERequestLocalUser = 0 ,
gdcm::network::eTransportConnConfirmLocal ,
gdcm::network::eASSOCIATE_ACPDUreceived ,
gdcm::network::eASSOCIATE_RJPDUreceived ,
gdcm::network::eTransportConnIndicLocal ,
gdcm::network::eAASSOCIATE_RQPDUreceived ,
gdcm::network::eAASSOCIATEResponseAccept ,
gdcm::network::eAASSOCIATEResponseReject ,
gdcm::network::ePDATArequest ,
gdcm::network::ePDATATFPDU ,
gdcm::network::eARELEASERequest ,
gdcm::network::eARELEASE_RQPDUReceivedOpen ,
gdcm::network::eARELEASE_RPPDUReceived ,
gdcm::network::eARELEASEResponse ,
gdcm::network::eAABORTRequest ,
gdcm::network::eAABORTPDUReceivedOpen ,
gdcm::network::eTransportConnectionClosed ,
gdcm::network::eARTIMTimerExpired ,
gdcm::network::eUnrecognizedPDUReceived ,
gdcm::network::eEventDoesNotExist }

Variables

- const int gdcm::network::cMaxEventID = eEventDoesNotExist

13.526 gdcmNetworkEvents.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018  /*
00019  The NetworkEvents enumeration defines the inputs into the state of the network connection.
00020
00021 These inputs can come either from user input or input from other things on the socket,
00022 ie, responses from the peer or ARTIM timeouts.
00023
00024 Note that this enumeration is not 'power of two', like the states, because you can't have
00025 multiple simultaneous events. Multiple state outputs in transition tables, however, is possible.
00026
00027 */

```

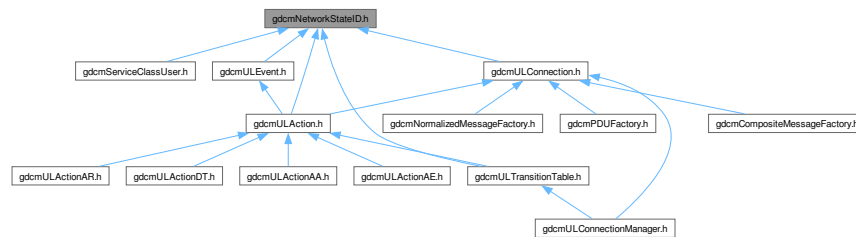
```

00028 #ifndef GDCMNETWORKEVENTS_H
00029 #define GDCMNETWORKEVENTS_H
00030
00031 namespace gdc {
00032     namespace network {
00033         typedef enum {
00034             eASSOCIATERequestLocalUser = 0,
00035             eTransportConnConfirmLocal,
00036             eASSOCIATE_ACPDUreceived,
00037             eASSOCIATE_RJPDUreceived,
00038             eTransportConnIndicLocal,
00039             eASSOCIATE_RQPDUreceived,
00040             eASSOCIATEResponseAccept,
00041             eASSOCIATEResponseReject,
00042             ePDATArequest,
00043             ePDATATFPDU,
00044             eARELEASERequest,
00045             eARELEASE_RQPDUReceivedOpen,
00046             eARELEASE_RPPDUReceived,
00047             eARELEASEResponse,
00048             eAABORTRequest,
00049             eAABORTPDUReceivedOpen,
00050             eTransportConnectionClosed,
00051             eARTIMTimerExpired,
00052             eUnrecognizedPDUReceived,
00053             eEventDoesNotExist
00054         } EEventID;
00055
00056         const int cMaxEventID = eEventDoesNotExist;
00057     }
00058 }
00059
00060 #endif //NETWORKEVENTS_H

```

13.527 gdcNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `gdc`
- namespace `gdc::network`

Enumerations

- enum `gdc::network::EStateID` {
`gdc::network::eStaDoesNotExist` = 0 ,
`gdc::network::eStaIdle` = 1 ,

```

gdcm::network::eSta2Open = 2 ,
gdcm::network::eSta3WaitLocalAssoc = 4 ,
gdcm::network::eSta4LocalAssocDone = 8 ,
gdcm::network::eSta5WaitRemoteAssoc = 16 ,
gdcm::network::eSta6TransferReady = 32 ,
gdcm::network::eSta7WaitRelease = 64 ,
gdcm::network::eSta8WaitLocalRelease = 128 ,
gdcm::network::eSta9ReleaseCollisionRqLocal = 256 ,
gdcm::network::eSta10ReleaseCollisionAc = 512 ,
gdcm::network::eSta11ReleaseCollisionRq = 1024 ,
gdcm::network::eSta12ReleaseCollisionAcLocal = 2048 ,
gdcm::network::eSta13AwaitingClose = 4096 }

```

Functions

- int `gdcm::network::GetStateIndex` (EStateID inState)

Variables

- const int `gdcm::network::cMaxStateID` = 13

13.528 gdcmNetworkStateID.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMNETWORKSTATEID_H
00019 #define GDCMNETWORKSTATEID_H
00020
00021 namespace gdcm {
00022     namespace network {
00023
00024         enum EStateID {
00025             eStaDoesNotExist = 0,
00026             eStaIdle = 1,
00027             eSta2Open = 2,
00028             eSta3WaitLocalAssoc = 4,
00029             eSta4LocalAssocDone = 8,
00030             eSta5WaitRemoteAssoc = 16,
00031             eSta6TransferReady = 32,
00032             eSta7WaitRelease = 64,
00033             eSta8WaitLocalRelease = 128,
00034             eSta9ReleaseCollisionRqLocal = 256,
00035             eSta10ReleaseCollisionAc = 512,
00036             eSta11ReleaseCollisionRq = 1024,

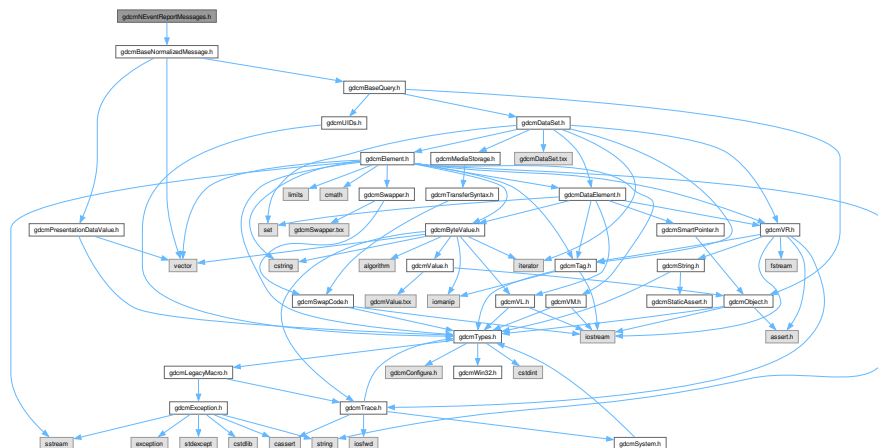
```

```
00089 #endif //GDCMNETWORKSTATEID_H
```

13.529 gdcMNEEventReportMessages.h File Reference

```
#include "gdcmBaseNormalizedMessage.h"
```

Include dependency graph for gdcMNEEventReportMessages.h:



Classes

- class `gdcm::network::NEventReportRQ`
`NEventReportRQ`.
- class `gdcm::network::NEventReportRSP`
`NEventReportRSP` this file defines the messages for the neventreport action.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.530 gdcmNEventReportMessages.h

[Go to the documentation of this file.](#)

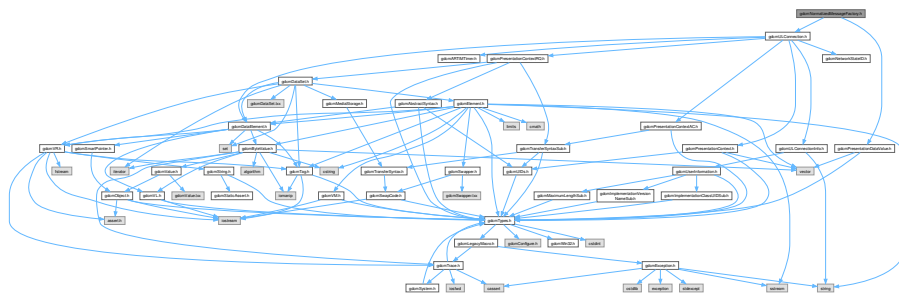
```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCNEVENTREPORTMESSAGES_H
00015 #define GDCMCNEVENTREPORTMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020     namespace network{
00021
00022     class ULConnection;
00023
00028     class NEventReportRQ : public BaseNormalizedMessage {
00029     public:
00030         std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031             const BaseQuery* inQuery) override;
00032     };
00033
00038     class NEventReportRSP : public BaseNormalizedMessage {
00039     public:
00040         std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041     };
00042     }
00043 }
00044 #endif // GDCMCNEVENTREPORTMESSAGES_H

```


13.533 gdcmmNormalizedMessageFactory.h File Reference

Include dependency graph for gdcMNormalizedMessageFactory.h:



- `class gdcn::network::NormalizedMessageFactory`

- namespace `gdcm`
- namespace `gdcm::network`

13.534 gdcmNormalizedMessageFactory.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMNORMALIZEDMESSAGEFACTORY_H
00015 #define GDCMNORMALIZEDMESSAGEFACTORY_H
00016
00017 #include "gdcmPresentationDataValue.h"
00018 #include "gdcmULConnection.h"
00019
00020 namespace gdcm {
00021     class BaseQuery;
00022     class File;
00023     namespace network {
00024         class BasePDU;
00025
00026     class NormalizedMessageFactory
00027     {
00028     public:
00029         static std::vector<PresentationDataValue> ConstructNEventReport (const ULConnection& inConnection, const
BaseQuery* inQuery);
00030         static std::vector<PresentationDataValue> ConstructNGet (const ULConnection& inConnection, const BaseQuery*
inQuery);
00031         static std::vector<PresentationDataValue> ConstructNSet (const ULConnection& inConnection, const BaseQuery*
inQuery);
00032         static std::vector<PresentationDataValue> ConstructNAction (const ULConnection& inConnection, const BaseQuery*
inQuery);
00033         static std::vector<PresentationDataValue> ConstructNCreate (const ULConnection& inConnection, const BaseQuery*
inQuery);
00034         static std::vector<PresentationDataValue> ConstructNDelete (const ULConnection& inConnection, const BaseQuery*
inQuery);
00035
00036     };
00037 }
00038 }
00039 }
00040
00041 #endif // GDCMNORMALIZEDMESSAGEFACTORY_H

```

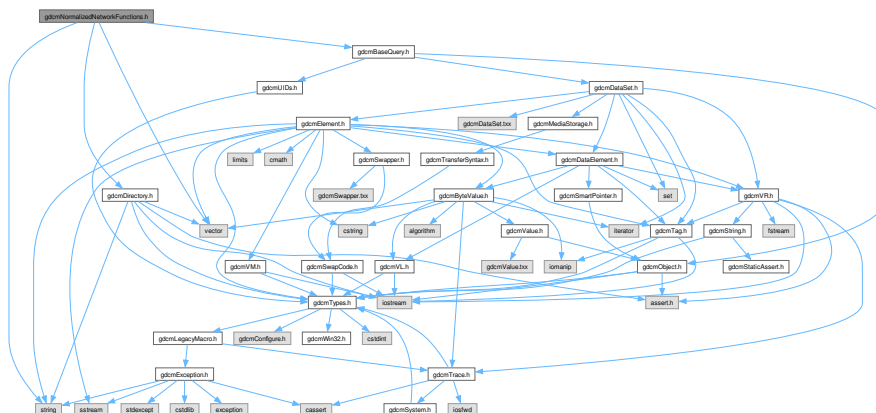
13.535 gdcmNormalizedNetworkFunctions.h File Reference

```

#include "gdcmDirectory.h"
#include "gdcmBaseQuery.h"
#include <vector>
#include <string>

```


Include dependency graph for gdcnNormalizedNetworkFunctions.h:



Classes

- class [gdcn::NormalizedNetworkFunctions](#)
Normalized Network Functions.

Namespaces

- namespace [gdcn](#)

13.536 gdcnNormalizedNetworkFunctions.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcn.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMNORMALIZEDNETWORKFUNCTIONS_H
00015  #define GDCMNORMALIZEDNETWORKFUNCTIONS_H
00016
00017  #include "gdcnDirectory.h"
00018  #include "gdcnBaseQuery.h" // EQueryLevel / EQueryType
00019
00020  #include <vector>
00021  #include <string>
00022
00023  namespace gdcn
00024  {
00046  class GDCM_EXPORT NormalizedNetworkFunctions
00047  {

```


Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.538 gdcmNSetMessages.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2014 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMCNSETMESSAGES_H
00015 #define GDCMCNSETMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020     namespace network{
00021
00022         class ULConnection;
00023
00028         class NSetRQ : public BaseNormalizedMessage {
00029         public:
00030             std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031             const BaseQuery* inQuery) override;
00032         };
00033
00038         class NSetRSP : public BaseNormalizedMessage {
00039         public:
00040             std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041         };
00042     }
00043 }
00044 #endif // GDCMCNSETMESSAGES_H

```

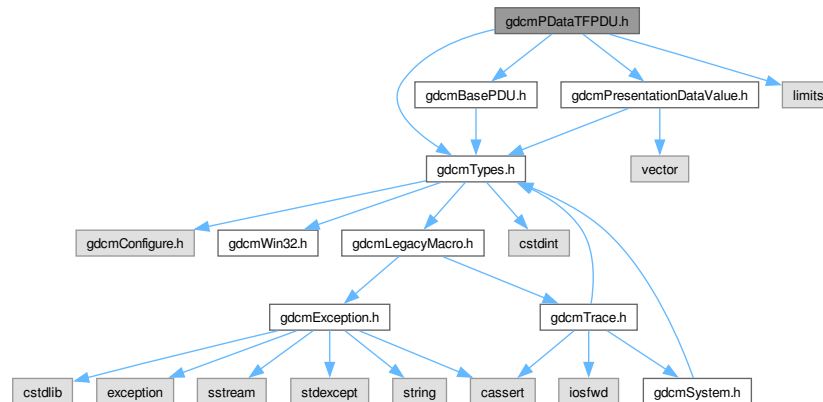
13.539 gdcmPDDataTFPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>

```

Include dependency graph for `gdcmPDataTFPDU.h`:



Classes

- class `gdcm::network::PDataTFPDU`
`PDataTFPDU`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.540 `gdcmPDataTFPDU.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMPDATATFPDU_H
00015  #define GDCMPDATATFPDU_H
00016  00017  #include "gdcmTypes.h"
00018  #include "gdcmPresentationDataValue.h"
00019  #include "gdcmBasePDU.h"
00020  #include <limits>
00021  00022  namespace gdcm

```

13.541 gdcMPDUFactory.h File Reference

Classes

- class [gdcmm::network::PDUFactory](#)
[PDUFactory](#) basically, given an initial byte, construct the.

Namespaces

- namespace [gdcmm](#)
- namespace [gdcmm::network](#)

13.542 gdcmmPDUFactory.h

[Go to the documentation of this file.](#)

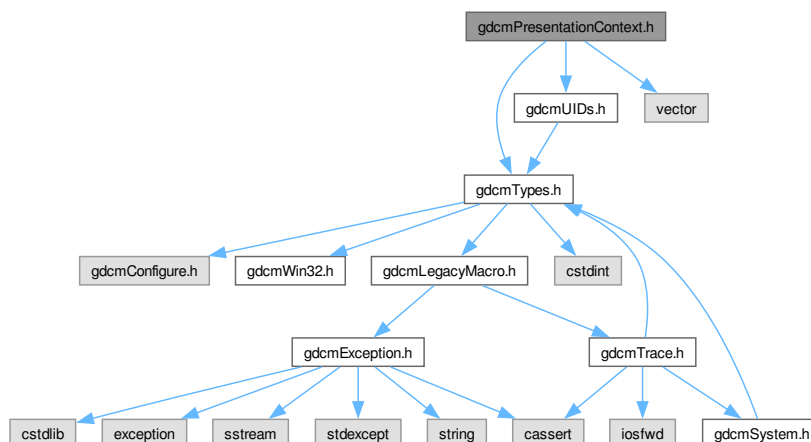
```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMPDUFACTORY_H
00019  #define GDCMPDUFACTORY_H
00020
00021  #include "gdcmmTypes.h"
00022  #include "gdcmmNetworkEvents.h"
00023  #include "gdcmmULConnection.h"
00024  #include "gdcmmPresentationDataValue.h"
00025
00026  namespace gdcmm{
00027  class BaseRootQuery;
00028  class BaseQuery;
00029  class File;
00030  namespace network{
00031  class BasePDU;
00032
00033  class PDUFactory {
00034  public:
00035  static BasePDU* ConstructPDU(uint8_t itemType); //eventually needs to be smartpointer'd
00036  static EEventID DetermineEventByPDU(const BasePDU* inPDU);
00037  static BasePDU* ConstructReleasePDU();
00038  static BasePDU* ConstructAbortPDU();
00039
00040  //these are the composite PDU construction methods for the PDataPDUs.
00041  //basically, builds a pdatapdu, and then puts the appropriate information in
00042  //for the appropriate composite service (c-echo, c-find, c-store, c-get, c-move)
00043  //the connection is necessary to construct the stream of PDVs that will
00044  //be then placed into the vector of PDUs
00045  static std::vector<BasePDU*> CreateCEchoPDU(const ULConnection& inConnection);
00046  static std::vector<BasePDU*> CreateCStoreRQPDU(const ULConnection& inConnection, const File &file, bool
writeDataSet = true);
00047  static std::vector<BasePDU*> CreateCStoreRSPDPDU(const DataSet *inDataSet, const BasePDU* inPC);
00048  static std::vector<BasePDU*> CreateCFindPDU(const ULConnection& inConnection, const BaseRootQuery*
inRootQuery);
00049  static std::vector<BasePDU*> CreateCMovePDU(const ULConnection& inConnection, const BaseRootQuery*
inRootQuery);
00050
00051  static std::vector<BasePDU*> CreateNEventReportPDU (const ULConnection& inConnection, const BaseQuery *inQuery);

```

13.543 gdcMPresentationContext.h File Reference

Include dependency graph for `gdcMPresentationContext.h`:



Classes

- class `gdcm::PresentationContext`
`PresentationContext`.

Namespaces

- namespace `gdcm`

13.544 `gdcmPresentationContext.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013  #ifndef GDCMPRESENTATIONCONTEXT_H
00014  #define GDCMPRESENTATIONCONTEXT_H
00015
00016  #include "gdcmTypes.h"
00017  #include "gdcmUIDs.h"
00018
00019  #include <vector>
00020
00021  namespace gdcm
00022  {
00023  {
00024
00025  class GDCM_EXPORT PresentationContext
00026  {
00027  public:
00028    PresentationContext();
00029
00030    PresentationContext( UIDs::TSName asname,
00031      UIDs::TSName tname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );
00032
00033    void SetAbstractSyntax( const char *absyn ) { AbstractSyntax = absyn; }
00034    const char *GetAbstractSyntax() const { return AbstractSyntax.c_str(); }
00035
00036    void AddTransferSyntax( const char *tsstr );
00037    typedef std::vector<std::string> TransferSyntaxArrayType;
00038    typedef TransferSyntaxArrayType::size_type SizeType;
00039    const char *GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i].c_str(); }
00040    SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }
00041
00042    void SetPresentationContextID( uint8_t id );
00043    uint8_t GetPresentationContextID() const;
00044
00045    void Print(std::ostream &os) const;
00046
00047    bool operator==(const PresentationContext & pc) const
00048    {
00049      gdcm_assert( TransferSyntaxes.size() == 1 ); // TODO
00050      gdcm_assert( pc.TransferSyntaxes.size() == 1 );
00051      return AbstractSyntax == pc.AbstractSyntax && TransferSyntaxes == pc.TransferSyntaxes;
00052    }
00053
00054  protected :
00055    std::string AbstractSyntax;

```



```

00063  std::vector<std::string> TransferSyntaxes;
00064  uint8_t /*PresentationContext*/ID;
00065  };
00066
00067  } // end namespace gdcm
00068
00069  #endif //GDCMPRESENTATIONCONTEXT_H

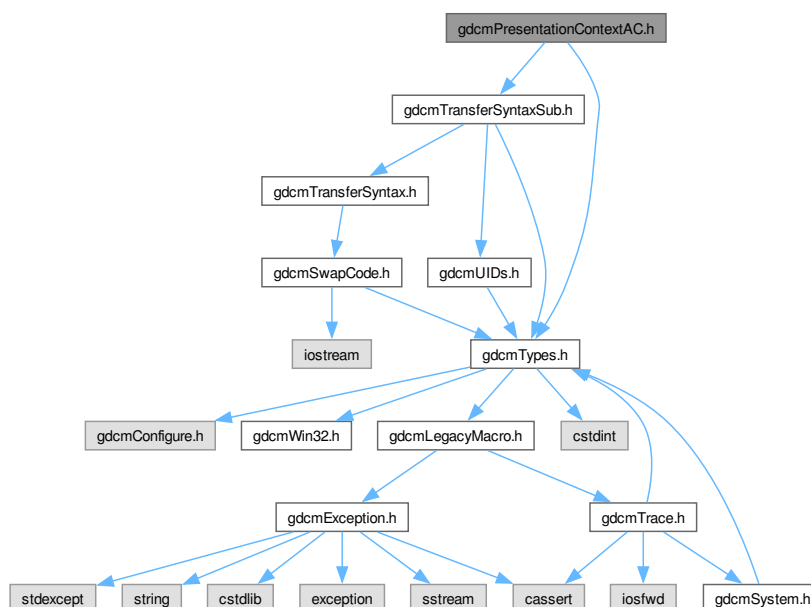
```

13.545 gdcmPresentationContextAC.h File Reference

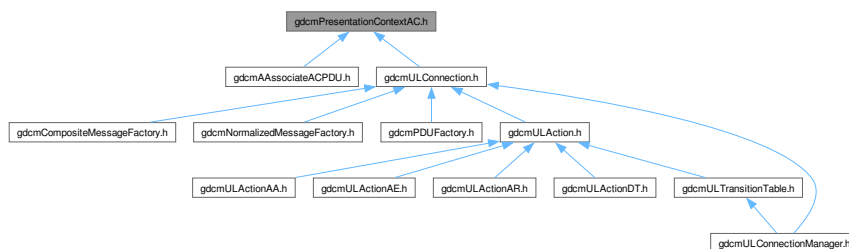
#include "gdcmTypes.h"

#include "gdcmTransferSyntaxSub.h"

Include dependency graph for gdcmPresentationContextAC.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::PresentationContextAC](#)
[PresentationContextAC](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.546 [gdcmPresentationContextAC.h](#)

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMPRESENTATIONCONTEXTAC_H
00015  #define GDCMPRESENTATIONCONTEXTAC_H
00016
00017  #include "gdcmTypes.h"
00018  #include "gdcmTransferSyntaxSub.h"
00019
00020  namespace gdcm
00021  {
00022
00023  namespace network
00024  {
00025
00032  class PresentationContextAC
00033  {
00034  public:
00035    PresentationContextAC();
00036    std::istream &Read(std::istream &is);
00037    const std::ostream &Write(std::ostream &os) const;
00038
00039    size_t Size() const;
00040
00041    void SetTransferSyntax( TransferSyntaxSub const &ts );
00042    void SetPresentationContextID( uint8_t id );
00043
00044    void Print(std::ostream &os) const;
00045
00046    uint8_t GetPresentationContextID() const
00047    {
00048      return ID;
00049    }
00050    TransferSyntaxSub const & GetTransferSyntax() const { return SubItems; }
00051
00052    void SetReason( uint8_t r ) { Result = r; }
00053    uint8_t GetReason() const { return Result; }
00054
00055  private:
00056    static const uint8_t ItemType;
00057    static const uint8_t Reserved2;
00058    uint16_t ItemLength; // len of last transfer syntax
00059    uint8_t /*PresentationContext*/ID;
00060    static const uint8_t Reserved6;

```

```

00061  uint8_t /*Reason*/Result;
00062  static const uint8_t Reserved8;
00063  TransferSyntaxSub SubItems;
00064  };
00065
00066  } // end namespace network
00067
00068  } // end namespace gdcm
00069
00070  #endif //GDCMPRESENTATIONCONTEXTAC_H

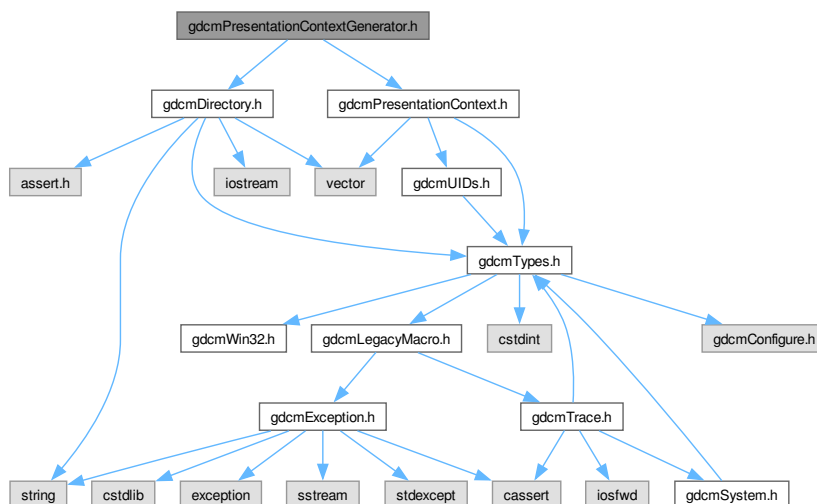
```

13.547 gdcmPresentationContextGenerator.h File Reference

#include "gdcmDirectory.h"

#include "gdcmPresentationContext.h"

Include dependency graph for gdcmPresentationContextGenerator.h:



Classes

- class `gdcm::PresentationContextGenerator`
`PresentationContextGenerator`.

Namespaces

- namespace `gdcm`

13.548 gdcmPresentationContextGenerator.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPRESENTATIONCONTEXTGENERATOR_H
00015 #define GDCMPRESENTATIONCONTEXTGENERATOR_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmPresentationContext.h"
00019
00020 namespace gdcm
00021 {
00022 class TransferSyntax;
00023 class File;
00024
00025 class GDCM_EXPORT PresentationContextGenerator
00026 {
00027 public:
00028     PresentationContextGenerator();
00029
00030     // Set MergeMode
00031     // Default mode, each pair AbstractSyntax/TransferSyntax are only merged when
00032     // exactly identical
00033     void SetMergeModeToAbstractSyntax();
00034
00035     // Set MergeMode
00036     // Merge is done on a per AbstractSyntax basis. Any new TransferSyntax for a
00037     // given AbstractSyntax is merge to the existing PresentationContext referring
00038     // to that AbstractSyntax
00039     void SetMergeModeToTransferSyntax();
00040
00041     bool GenerateFromUID(UIDs::TSName asname);
00042
00043     bool GenerateFromFilenames(const Directory::FilenamesType &files);
00044
00045     bool AddFromFile(const File &file);
00046
00047     typedef std::vector<PresentationContext> PresentationContextArrayType;
00048     typedef PresentationContextArrayType::size_type SizeType;
00049     PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }
00050
00051     void SetDefaultTransferSyntax( const TransferSyntax &ts );
00052 protected:
00053     bool AddPresentationContext( const char *absyn, const char *ts );
00054     const char *GetDefaultTransferSyntax() const;
00055 private:
00056     std::vector<PresentationContext> PresContext;
00057     static std::string DefaultTransferSyntax;
00058 };
00059
00060 } // end namespace gdcm
00061
00062 #endif //GDCMPRESENTATIONCONTEXTGENERATOR_H

```

13.549 gdcmPresentationContextRQ.h File Reference

```

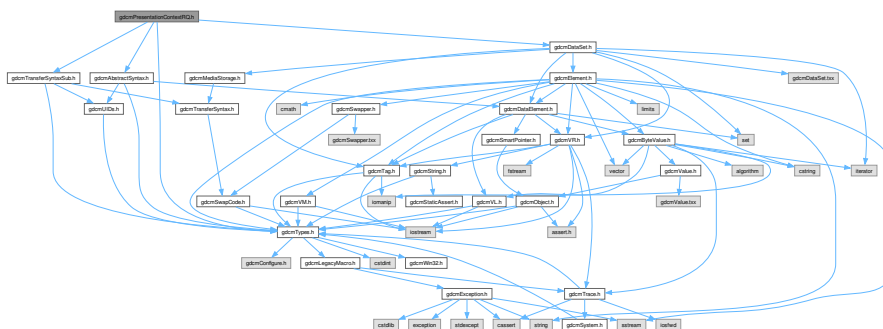
#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"

```

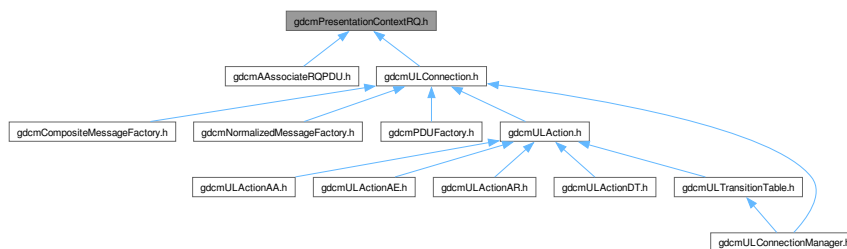
```
#include "gdcmpresentationcontextrq.h"
```

```
#include "gdcmdataset.h"
```

Include dependency graph for gdcmpresentationcontextrq.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::network::PresentationContextRQ`
`PresentationContextRQ`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.550 gdcmpresentationcontextrq.h

[Go to the documentation of this file.](#)

```
00001
```

```
/*=====
```

```
00002
```

```
00003 Program: GDCM (Grassroots DICOM). A DICOM library
```

```
00004
```

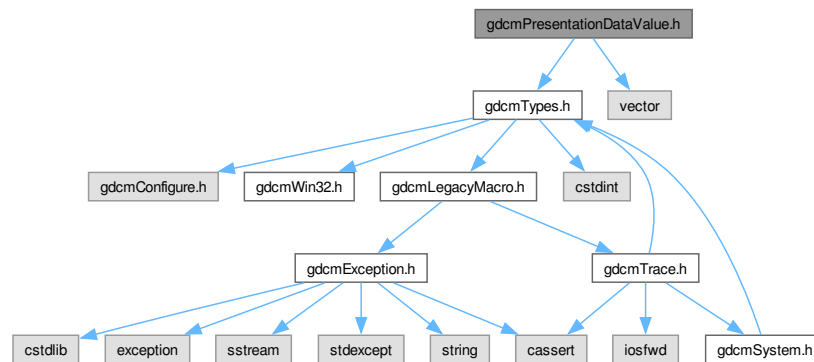
```

00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011 PURPOSE. See the above copyright notice for more information.
00012
00013
00014 =====*/
00014 #ifndef GDCMPRESENTATIONCONTEXTTRQ_H
00015 #define GDCMPRESENTATIONCONTEXTTRQ_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmAbstractSyntax.h"
00019 #include "gdcmTransferSyntaxSub.h"
00020 #include "gdcmDataSet.h"
00021
00022 namespace gdcm
00023 {
00024 class PresentationContext;
00025 namespace network
00026 {
00027
00034 class GDCM_EXPORT PresentationContextRQ
00035 {
00036 public:
00037 PresentationContextRQ();
00038
00042 PresentationContextRQ( UIDs::TSName asname, UIDs::TSName tsname =
00043 UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );
00044
00045 std::istream &Read(std::istream &is);
00046 const std::ostream &Write(std::ostream &os) const;
00047 size_t Size() const;
00048
00049 void SetAbstractSyntax( AbstractSyntax const &absyn );
00050 AbstractSyntax const &GetAbstractSyntax() const { return SubItems; }
00051 AbstractSyntax &GetAbstractSyntax() { return SubItems; }
00052
00053 void AddTransferSyntax( TransferSyntaxSub const &ts );
00054 typedef std::vector<TransferSyntaxSub>::size_type SizeType;
00055 TransferSyntaxSub const & GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i]; }
00056 TransferSyntaxSub & GetTransferSyntax(SizeType i) { return TransferSyntaxes[i]; }
00057 std::vector<TransferSyntaxSub> const & GetTransferSyntaxes() const { return TransferSyntaxes; }
00058 SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }
00059
00060 void SetPresentationContextID( uint8_t id );
00061 uint8_t GetPresentationContextID() const;
00062
00063 void Print(std::ostream &os) const;
00064
00065 bool operator==(const PresentationContextRQ & pc) const
00066 {
00067 gdcm_assert( TransferSyntaxes.size() == 1 ); // TODO
00068 gdcm_assert( pc.TransferSyntaxes.size() == 1 );
00069 return SubItems == pc.SubItems && TransferSyntaxes == pc.TransferSyntaxes;
00070 }
00071
00072 PresentationContextRQ(const PresentationContext & pc);
00073
00074 private:
00075 static const uint8_t ItemType;
00076 static const uint8_t Reserved2;
00077 uint16_t ItemLength; // len of last transfer syntax
00078 uint8_t /*PresentationContext*/ID;
00079 static const uint8_t Reserved6;
00080 static const uint8_t Reserved7;
00081 static const uint8_t Reserved8;
00082 /*
00083 This variable field shall contain the following sub-items: one Abstract
00084 Syntax and one or more Transfer Syntax(es). For a complete
00085 description of the use and encoding of these sub-items see Sections
00086 9.3.2.2.1 and 9.3.2.2.2.
00087 */
00088 AbstractSyntax SubItems;
00089 std::vector<TransferSyntaxSub> TransferSyntaxes;
00090 };
00091
00092 } // end namespace network
00093

```

13.551 gdcmPresentationDataValue.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::network::PresentationDataValue`
`PresentationDataValue`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.552 gdcmPresentationDataValue.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMPRESENTATIONDATAVALUE_H
00015 #define GDCMPRESENTATIONDATAVALUE_H
00016
00017 #include "gdcmTypes.h"
00018
00019 #include <vector>
00020
00021 namespace gdcm
00022 {
00023 class DataSet;
00024 namespace network
00025 {
00026
00032 class GDCM_EXPORT PresentationDataValue
00033 {
00034 public:
00035   PresentationDataValue();
00036   std::istream &Read(std::istream &is);
00037   std::istream &ReadInto(std::istream &is, std::ostream &os);
00038
00039   const std::ostream &Write(std::ostream &os) const;
00040
00042   size_t Size() const;
00043
00046   void SetDataSet(const DataSet & ds);
00047   void SetBlob(const std::string & partialblob);
00048   const std::string &GetBlob() const;
00049
00050   uint8_t GetPresentationContextID() const { return PresentationContextID; }
00051   void SetPresentationContextID(uint8_t id) {
00052     gdcm_assert( id );
00053     PresentationContextID = id;
00054   }
00055   uint8_t GetMessageHeader() const {
00056     gdcm_assert( MessageHeader <= 0x3 );
00057     return MessageHeader;
00058   }
00059   // E.2 MESSAGE CONTROL HEADER ENCODING
00060   // Only the first two bits are considered
00061   void SetMessageHeader(uint8_t messageheader) {
00062     MessageHeader = messageheader;
00063     gdcm_assert( MessageHeader <= 0x3 );
00064   }
00065   //flip the least significant bit of the message header to 1
00066   //if this is a command, else set it to 0.
00067   void SetCommand(bool inCommand);
00068   void SetLastFragment(bool inLast); //set to true if this is the last PDV of a set
00069
00070   bool GetIsCommand() const;
00071   bool GetIsLastFragment() const;
00072
00073   void Print(std::ostream &os) const;
00074
00075   //NOTE that the PDVs have to be given in the order in which they were received!
00076   //also note that a dataset may be across multiple PDVs
00077   static DataSet ConcatenatePDVBlobs(const std::vector<PresentationDataValue>& inPDVs);
00078
00079   static DataSet ConcatenatePDVBlobsAsExplicit(const std::vector<PresentationDataValue>& inPDVs);
00080
00081 private:
00082

```



```

00083  uint32_t ItemLength;
00084  uint8_t PresentationContextID;
00085  uint8_t MessageHeader;
00086  std::string Blob;
00087  };
00088  } // end namespace network
00089  } // end namespace gdcm
00091
00092  #endif //GDCMPRESENTATIONDATAVALUE_H

```

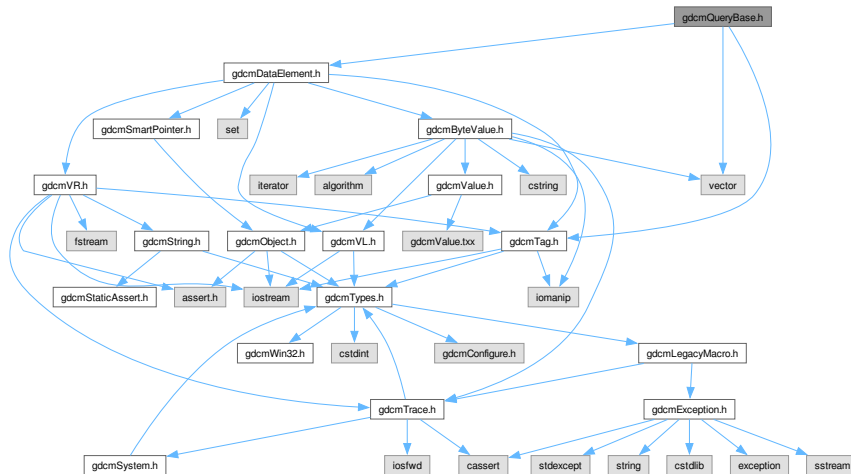
13.553 gdcmQueryBase.h File Reference

```
#include "gdcmTag.h"
```

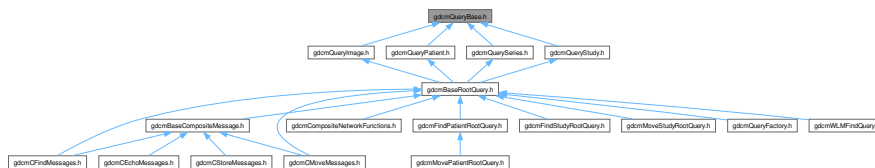
```
#include "gdcmDataElement.h"
```

```
#include <vector>
```

Include dependency graph for gdcmQueryBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::QueryBase](#)
QueryBase.

Namespaces

- namespace [gdcm](#)

Enumerations

- enum [gdcm::ERootType](#) {
[gdcm::ePatientRootType](#) ,
[gdcm::eStudyRootType](#) }

13.554 [gdcmQueryBase.h](#)

[Go to the documentation of this file.](#)

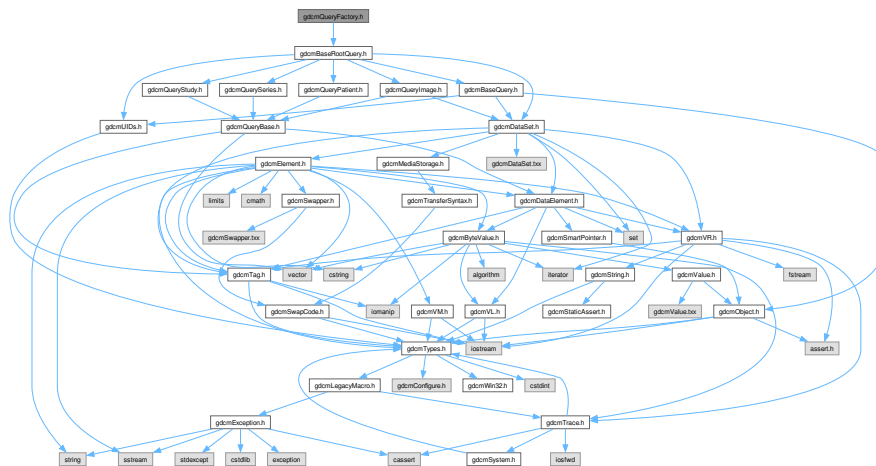
```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYBASE_H
00019 #define GDCMQUERYBASE_H
00020
00021 #include "gdcmTag.h"
00022 #include "gdcmDataElement.h"
00023
00024 #include <vector>
00025
00026 namespace gdcm
00027 {
00028     enum ERootType
00029     {
00030         ePatientRootType,
00031         eStudyRootType
00032     };
00033
00060 class GDCM_EXPORT QueryBase
00061 {
00062 public:
00063     virtual ~QueryBase() = default;
00064
00065     virtual std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const = 0;
00066     virtual std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const = 0;
00067     virtual std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const = 0;
00068     // C.4.1.2.1 Baseline Behavior of SCU
00069     // All C-FIND SCUs shall be capable of generating query requests which
00070     // meet the requirements of the Hierarchical Search.
00071     // The Identifier contained in a C-FIND request shall contain a single
00072     // value in the Unique Key Attribute for each level above the
00073     // Query/Retrieve level. No Required or Optional Keys shall be
00074     // specified which are associated with levels above the Query/Retrieve
00075     // level.
00077     virtual std::vector<Tag> GetHierarchicalSearchTags(const ERootType& inRootType) const = 0;
00078
00081     std::vector<Tag> GetAllTags(const ERootType& inRootType) const;
00082
00085     std::vector<Tag> GetAllRequiredTags(const ERootType& inRootType) const;

```

13.555 gdcMQueryFactory.h File Reference

Include dependency graph for `gdcMQueryFactory.h`:



- class `gdc::QueryFactory`
QueryFactory.h.

- namespace `gdcm`

- enum `gdcm::ECharSet` {
`gdcm::eLatin1` = 0 ,
`gdcm::eLatin2` ,
`gdcm::eLatin3` ,
`gdcm::eLatin4` ,
`gdcm::eCyrillic` ,
`gdcm::eArabic` ,
`gdcm::eGreek` ,

```

gdcM::eHebrew ,
gdcM::eLatin5 ,
gdcM::eJapanese ,
gdcM::eThai ,
gdcM::eJapaneseKanjiMultibyte ,
gdcM::eJapaneseSupplementaryKanjiMultibyte ,
gdcM::eKoreanHangulHanjaMultibyte ,
gdcM::eUTF8 ,
gdcM::eGB18030 }

```

13.556 gdcMQueryFactory.h

[Go to the documentation of this file.](#)

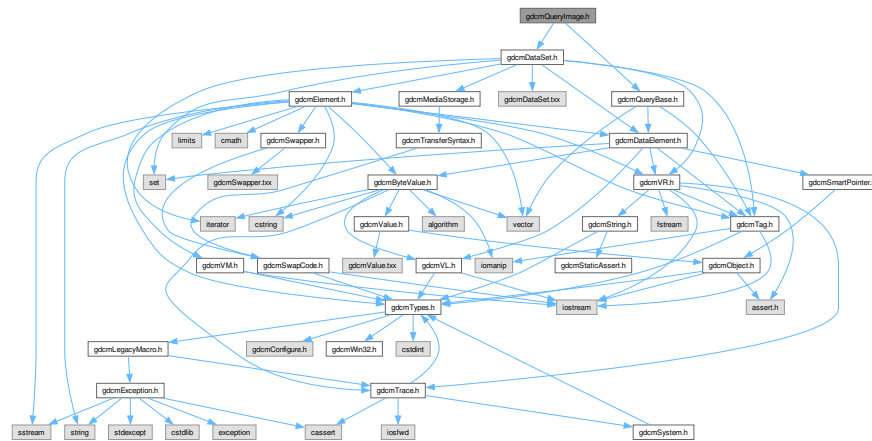
```

00001
00002  /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYFACTORY_H
00019 #define GDCMQUERYFACTORY_H
00020
00021 #include "gdcMBaseRootQuery.h"
00022
00023 namespace gdcM{
00024     enum ECharSet {
00025         eLatin1 = 0,
00026         eLatin2,
00027         eLatin3,
00028         eLatin4,
00029         eCyrillic,
00030         eArabic,
00031         eGreek,
00032         eHebrew,
00033         eLatin5, // Latin Alphabet No. 5 (Turkish) Extended
00034         eJapanese, // JIS X 0201 (Shift JIS) Extended
00035         eThai, // TIS 620-2533 (Thai) Extended
00036         eJapaneseKanjiMultibyte, // JIS X 0208 (Kanji) Extended
00037         eJapaneseSupplementaryKanjiMultibyte, // JIS X 0212 (Kanji) Extended
00038         eKoreanHangulHanjaMultibyte, // KS X 1001 (Hangul and Hanja) Extended
00039         eUTF8,
00040         eGB18030 // Chinese (Simplified) Extended
00041     };
00042
00043     class GDCM_EXPORT QueryFactory
00044     {
00045     public:
00046
00047         static BaseQuery* ProduceQuery( const std::string & sopInstanceUID, ENQueryType inQueryType );
00048         static BaseRootQuery* ProduceQuery(ERootType inRootType, EQueryType inQueryType,
00049             EQueryLevel inQueryLevel);
00050
00051         static DataElement ProduceCharacterSetDataElement(
00052             const std::vector<ECharSet>& inCharSetType);
00053
00054         static ECharSet GetCharacterFromCurrentLocale();
00055
00056         static void ListCharSets(std::ostream& os);

```

```
00086 };
00087
00088 } // end namespace gdcm
00089
00090 #endif // GDCMQUERYFACTORY_H
```

```
#include "gdcQueryBase.h"
#include "gdcDataSet.h"
Include dependency graph for gdcQueryImage.h:
```



```

graph TD
    glibmmQueryImage.h --> glibmmBasePoolQuery.h
    glibmmBasePoolQuery.h --> glibmmBaseCompositMessage.h
    glibmmBasePoolQuery.h --> glibmmComposeNetworkFunctions.h
    glibmmBasePoolQuery.h --> glibmmFindPatternQuery.h
    glibmmBasePoolQuery.h --> glibmmFindStudyQuery.h
    glibmmBasePoolQuery.h --> glibmmMoveStudyPoolQuery.h
    glibmmBasePoolQuery.h --> glibmmQueryFactory.h
    glibmmBasePoolQuery.h --> glibmmWM_MFQuery.h
    glibmmBaseCompositMessage.h --> glibmmCFindMessages.h
    glibmmBaseCompositMessage.h --> glibmmCFindMessages.h
    glibmmBaseCompositMessage.h --> glibmmCFindMessages.h
    glibmmBaseCompositMessage.h --> glibmmCFindMessages.h
    glibmmBaseCompositMessage.h --> glibmmMoveMessages.h
    glibmmBaseCompositMessage.h --> glibmmMovePatternQuery.h
  
```

- `class gdcm::QueryImage`
`QueryImage.`

- namespace `gdcm`

13.558 gdcmQueryImage.h

[Go to the documentation of this file.](#)

```

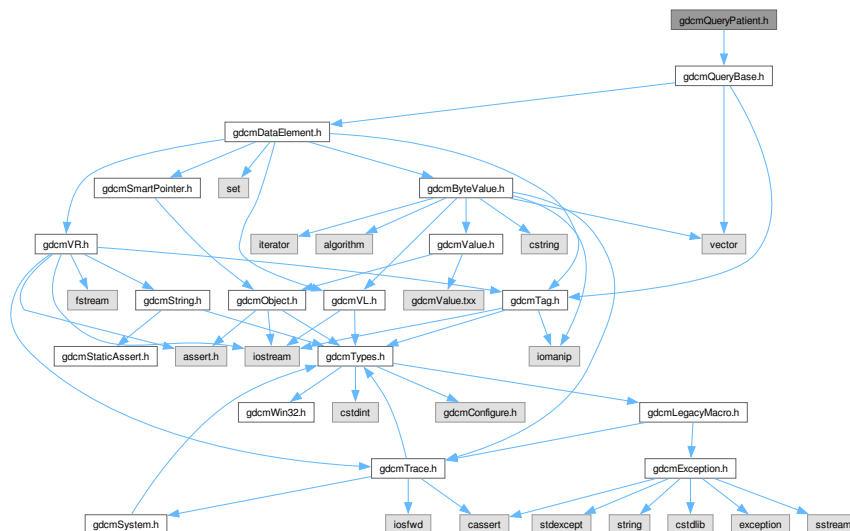
00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYIMAGE_H
00019 #define GDCMQUERYIMAGE_H
00020
00021 #include "gdcmQueryBase.h"
00022 #include "gdcmDataSet.h"
00023
00024 namespace gdcm
00025 {
00030 class GDCM_EXPORT QueryImage : public QueryBase
00031 {
00032 public:
00033     std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00034     std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00035     std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00036     std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
00037
00038     const char * GetName() const override;
00039
00040     DataElement GetQueryLevel() const override;
00041 };
00042
00043 } // end namespace gdcm
00044
00045 #endif // GDCMQUERYIMAGE_H

```

13.559 gdcmQueryPatient.h File Reference

#include "gdcmQueryBase.h"

Include dependency graph for gdcmQueryPatient.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::QueryPatient](#)
[QueryPatient](#).

Namespaces

- namespace [gdcm](#)

13.560 gdcmQueryPatient.h

[Go to the documentation of this file.](#)

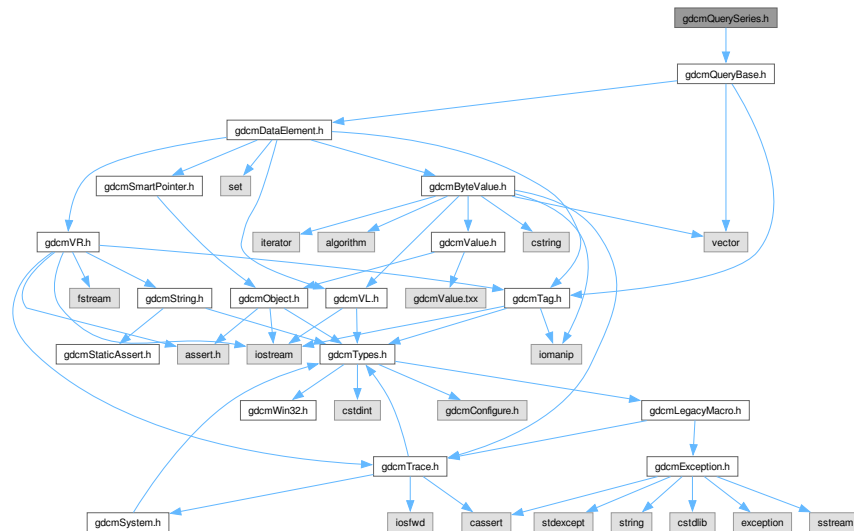
```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYPATIENT_H
00019 #define GDCMQUERYPATIENT_H
00020
00021 #include "gdcmQueryBase.h"
00022
00023 namespace gdcm
00024 {
00029 class GDCM_EXPORT QueryPatient : public QueryBase
00030 {
00031 public:
00032     std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00033     std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00034     std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00035     std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
00036
00037     const char * GetName() const override;
00038     DataElement GetQueryLevel() const override;
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMQUERYPATIENT_H

```



```
#include "gdcQueryBase.h"
// Include dependency graph for gdcQuerySeries.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::QuerySeries`
`QuerySeries`.

Namespaces

- namespace `gdcm`

13.562 gdcmQuerySeries.h

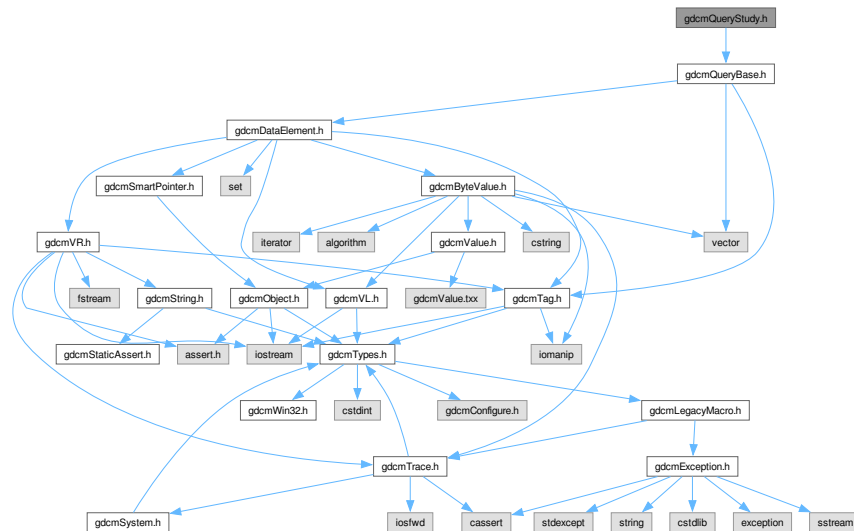
[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMQUERYSERIES_H
00019 #define GDCMQUERYSERIES_H
00020
00021 #include "gdcmQueryBase.h"
00022
00023 namespace gdcm
00024 {
00029 class GDCM_EXPORT QuerySeries : public QueryBase
00030 {
00031 public:
00032     std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00033     std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00034     std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00035     std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
00036
00037     const char * GetName() const override;
00038     DataElement GetQueryLevel() const override;
00039 };
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMQUERYSERIES_H

```

```
#include "gdcmQueryBase.h"
Include dependency graph for gdcmQueryStudy.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::QueryStudy`
QueryStudy.h.

Namespaces

- namespace `gdcm`

13.564 gdcmQueryStudy.h

[Go to the documentation of this file.](#)

```

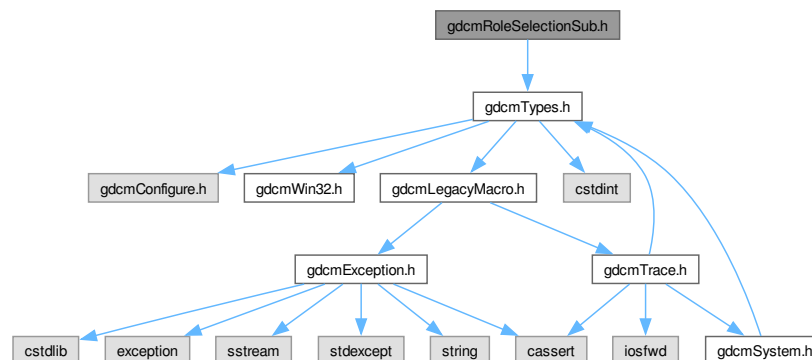
00001  /*=====
00002  *
00003  *   Copyright NumFOCUS
00004  *
00005  *   Licensed under the Apache License, Version 2.0 (the "License");
00006  *   you may not use this file except in compliance with the License.
00007  *   You may obtain a copy of the License at
00008  *
00009  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *   Unless required by applicable law or agreed to in writing, software
00012  *   distributed under the License is distributed on an "AS IS" BASIS,
00013  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *   See the License for the specific language governing permissions and
00015  *   limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMQUERYSTUDY_H
00019  #define GDCMQUERYSTUDY_H
00020
00021  #include "gdcmQueryBase.h"
00022
00023  namespace gdcm
00024  {
00029  class GDCM_EXPORT QueryStudy : public QueryBase
00030  {
00031  public:
00032      std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
00033      std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
00034      std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
00035      std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;
00036
00037      const char *GetName() const override;
00038      DataElement GetQueryLevel() const override;
00039  };
00040
00041  } // end namespace gdcm
00042
00043  #endif //GDCMQUERYSTUDY_H

```

13.565 gdcmRoleSelectionSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class [gdcm::network::RoleSelectionSub](#)
[RoleSelectionSub](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.566 gdcmRoleSelectionSub.h

[Go to the documentation of this file.](#)

```

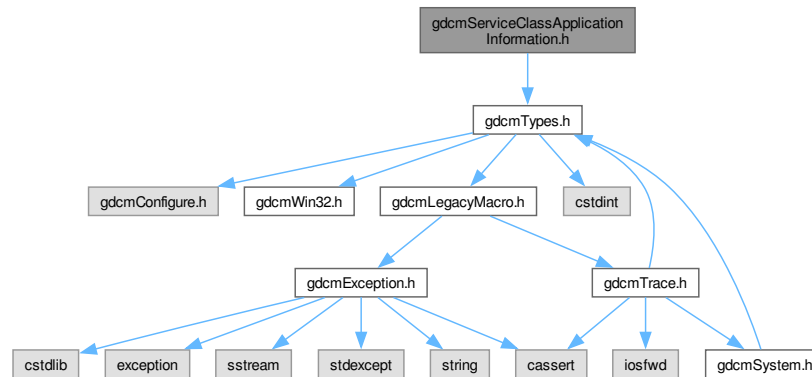
00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMROLESELECTIONSUB_H
00015  #define GDCMROLESELECTIONSUB_H
00016
00017  #include "gdcmTypes.h"
00018
00019  namespace gdcm
00020  {
00021
00022  namespace network
00023  {
00024
00025  class RoleSelectionSub
00026  {
00027  public:
00028  RoleSelectionSub();
00029  std::istream &Read(std::istream &is);
00030  const std::ostream &Write(std::ostream &os) const;
00031
00032  size_t Size() const;
00033  void Print(std::ostream &os) const;
00034
00035  void SetTuple(const char *uid, uint8_t scurole, uint8_t scpole);
00036
00037  private:
00038  static const uint8_t ItemType;
00039  static const uint8_t Reserved2;
00040  uint16_t ItemLength;
00041  uint16_t UIDLength;
00042  std::string /*SOP-class-uid*/ Name; // UID
00043  uint8_t SCURole;
00044  uint8_t SCPRole;
00045  };
00046  } // end namespace network
00047  } // end namespace gdcm
00048  #endif // GDCMROLESELECTIONSUB_H

```

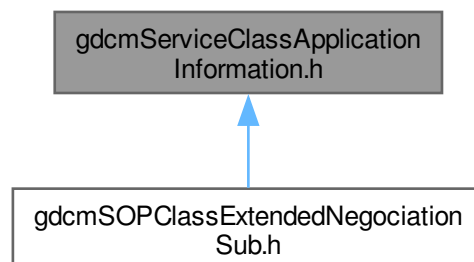
13.567 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ServiceClassApplicationInformation](#)

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.568 gdcmServiceClassApplicationInformation.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  #ifndef GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00015  #define GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00016  00017  #include "gdcmTypes.h"
00018  00019  namespace gdcm
00020  {
00021  00022  namespace network
00023  {
00024  00030  class ServiceClassApplicationInformation
00031  {
00032  public:
00033  ServiceClassApplicationInformation();
00034  std::istream &Read(std::istream &is);
00035  const std::ostream &Write(std::ostream &os) const;
00036  00037  size_t Size() const;
00038  void SetTuple(uint8_t levelofsupport, uint8_t levelofdigitalsig,
00039  uint8_t elementcoercion);
00040  00041  void Print(std::ostream &os) const;
00042  private:
00043  uint8_t InternalArray[6];
00044  };
00045  00046  } // end namespace network
00047  00048  } // end namespace gdcm
00049  00050  #endif //GDCMSERVICECLASSAPPLICATIONINFORMATION_H

```

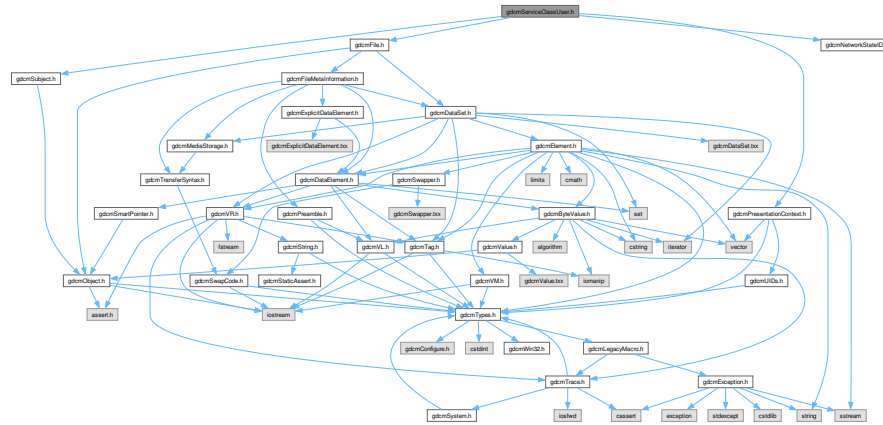
13.569 gdcmServiceClassUser.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
#include "gdcmFile.h"
#include "gdcmNetworkStateID.h"

```

Include dependency graph for `gdcmServiceClassUser.h`:



Classes

- class `gdcm::ServiceClassUser`
[ServiceClassUser](#).

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.570 gdcmServiceClassUser.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  #ifndef GDCMSERVICECLASSUSER_H
00015  #define GDCMSERVICECLASSUSER_H
00016
00017  #include "gdcmSubject.h"
00018
00019  #include "gdcmPresentationContext.h"
00020  #include "gdcmFile.h"
00021
00022  #include "gdcmNetworkStateID.h" // EStateID
00023
00024  namespace gdcm

```



```

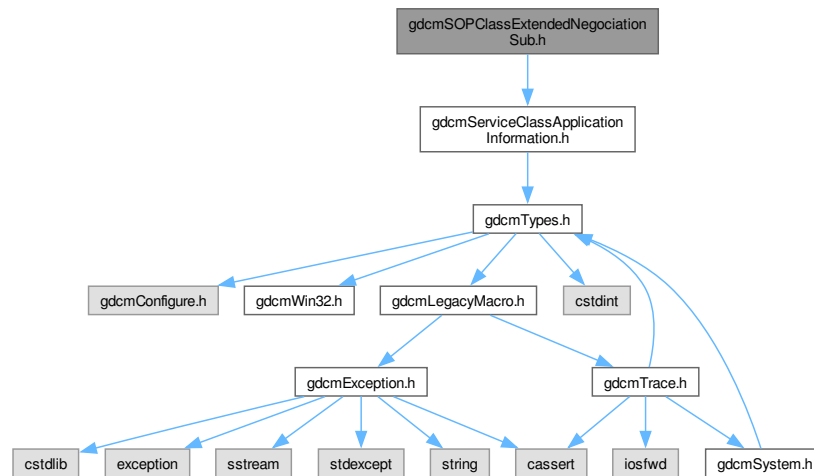
00025 {
00026 class ServiceClassUserInternals;
00027 class BaseRootQuery;
00028 namespace network{
00029 class ULEvent;
00030 class ULConnection;
00031 class ULConnectionCallback;
00032 }
00036 class GDCM_EXPORT ServiceClassUser : public Subject
00037 {
00038 public:
00042 ServiceClassUser();
00043 ~ServiceClassUser() override;
00044 ServiceClassUser(const ServiceClassUser&) = delete;
00045 void operator=(const ServiceClassUser &) = delete;
00046
00048 void SetHostname( const char *hostname );
00049
00051 void SetPort( uint16_t port );
00052
00054 void SetPortSCP( uint16_t portscp );
00055
00057 void SetAETitle(const char *aetitle);
00058 const char *GetAETitle() const;
00059
00061 void SetCalledAETitle(const char *aetitle);
00062 const char *GetCalledAETitle() const;
00063
00065 void SetTimeout(double t);
00066 double GetTimeout() const;
00067
00071 bool InitializeConnection();
00072
00074 void SetPresentationContexts(std::vector<PresentationContext> const & pcs);
00075
00077 bool IsPresentationContextAccepted(const PresentationContext& pc) const;
00078
00080 bool StartAssociation();
00081
00083 bool StopAssociation();
00084
00086 bool SendEcho();
00087
00089 bool SendStore(const char *filename);
00092 bool SendStore(File const &file);
00094 bool SendStore(DataSet const &ds);
00095
00097 bool SendFind(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
00098
00100 bool SendMove(const BaseRootQuery* query, const char *outputdir);
00102 bool SendMove(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
00104 bool SendMove(const BaseRootQuery* query, std::vector<File> &retFile);
00105
00107 static SmartPointer<ServiceClassUser> New() { return new ServiceClassUser; }
00108
00109 private:
00110 network::EStateID RunEventLoop(network::ULEvent& inEvent,
00111     network::ULConnection* inWhichConnection,
00112     network::ULConnectionCallback* inCallback, const bool& startWaiting);
00113 network::EStateID RunMoveEventLoop(network::ULEvent& inEvent,
00114     network::ULConnectionCallback* inCallback);
00115
00116 private:
00117 ServiceClassUserInternals *Internals;
00118 };
00119
00120 } // end namespace gdcm
00121
00122 #endif // GDCMSERVICECLASSUSER_H

```

13.571 gdcmSOPClassExtendedNegociationSub.h File Reference

#include "gdcmServiceClassApplicationInformation.h"

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class [gdcm::network::SOPClassExtendedNegociationSub](#)
[SOPClassExtendedNegociationSub](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.572 gdcmSOPClassExtendedNegociationSub.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/

```

```

00014 #ifndef GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
00015 #define GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
00016
00017 #include "gdcmServiceClassApplicationInformation.h"
00018
00019 namespace gdcm
00020 {
00021     namespace network
00022     {
00023
00031         class SOPClassExtendedNegociationSub
00032         {
00033         public:
00034             SOPClassExtendedNegociationSub();
00035             std::istream &Read(std::istream &is);
00036             const std::ostream &Write(std::ostream &os) const;
00037
00038             size_t Size() const;
00039             void Print(std::ostream &os) const;
00040
00041             void SetTuple(const char *uid, uint8_t levelofsupport = 3,
00042                 uint8_t levelofdigitalsig = 0,
00043                 uint8_t elementcoercion = 2);
00044
00045         private:
00046             static const uint8_t ItemType;
00047             static const uint8_t Reserved2;
00048             uint16_t ItemLength;
00049             uint16_t UIDLength;
00050             std::string /*SOP-class-uid*/ Name; // UID
00051             ServiceClassApplicationInformation SCAI;
00052         };
00053
00054     } // end namespace network
00055
00056 } // end namespace gdcm
00057
00058 #endif // GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H

```

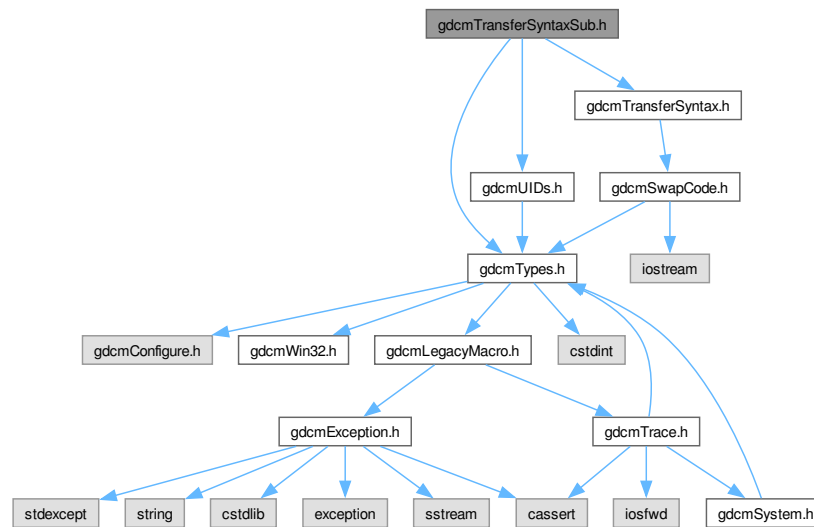
13.573 gdcmTransferSyntaxSub.h File Reference

```

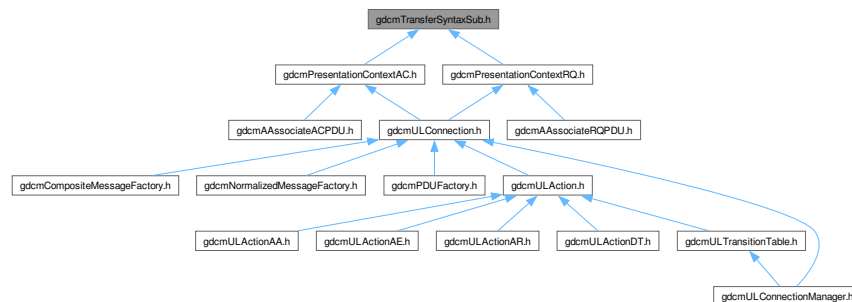
#include "gdcmTypes.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDs.h"

```

Include dependency graph for `gdcmTransferSyntaxSub.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::network::TransferSyntaxSub`
`TransferSyntaxSub`.

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.574 gdcmTransferSyntaxSub.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 #ifndef GDCMTRANSFERSYNTAXSUB_H
00015 #define GDCMTRANSFERSYNTAXSUB_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmTransferSyntax.h"
00019 #include "gdcmUIDs.h"
00020
00021 namespace gdcm
00022 {
00023
00024 namespace network
00025 {
00026
00037 class TransferSyntaxSub
00038 {
00039 public:
00040   TransferSyntaxSub();
00041   void SetName( const char *name );
00042   const char *GetName() const { return Name.c_str(); }
00043
00044   // accept a UIDs::TSType also...
00045   void SetNameFromUID( UIDs::TSName tsname );
00046
00047   std::istream &Read(std::istream &is);
00048   const std::ostream &Write(std::ostream &os) const;
00049   size_t Size() const;
00050   void Print(std::ostream &os) const;
00051
00052   bool operator==(const TransferSyntaxSub & ts) const
00053   {
00054     return Name == ts.Name;
00055   }
00056
00057 private:
00058   void UpdateName( const char *name );
00059   static const uint8_t ItemType;
00060   static const uint8_t Reserved2;
00061   uint16_t ItemLength; // len of
00062   std::string /*"TransferSyntaxSub"*/ Name; // UID
00063 };
00064
00065 } // end namespace network
00066
00067 } // end namespace gdcm
00068
00069 #endif //GDCMTRANSFERSYNTAXSUB_H

```

13.575 gdcmULAction.h File Reference

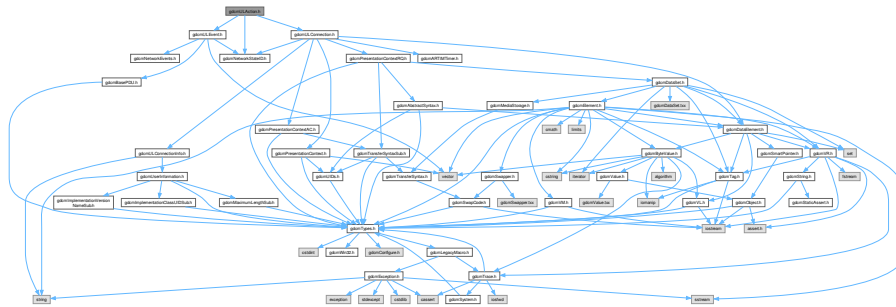
```

#include "gdcmNetworkStateID.h"
#include "gdcmULEvent.h"

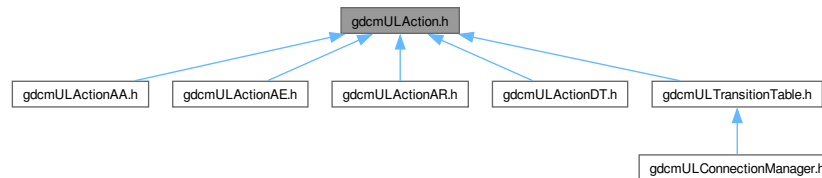
```

```
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmULAction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ULAction](#)
[ULAction](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.576 gdcmULAction.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  *
00004  * Copyright NumFOCUS
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
```

```

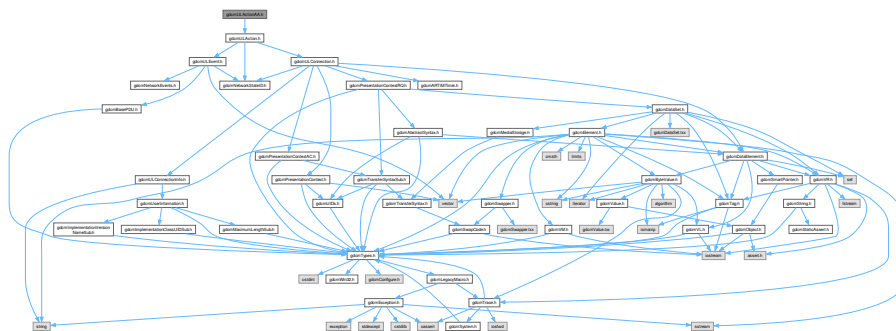
00010 *
00011 * Unless required by applicable law or agreed to in writing, software
00012 * distributed under the License is distributed on an "AS IS" BASIS,
00013 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014 * See the License for the specific language governing permissions and
00015 * limitations under the License.
00016 *
00017 *=====*/
00018 #ifndef GDCMULACTION_H
00019 #define GDCMULACTION_H
00020
00021 #include "gdcmNetworkStateID.h"
00022 #include "gdcmULEvent.h"
00023 #include "gdcmULConnection.h"
00024
00025 namespace gdcm {
00026 class Subject;
00027 namespace network {
00028
00062 class ULAction {
00063
00064 protected:
00065
00066
00067 public:
00068     ULAction() = default;
00069     //make sure destructors are virtual to avoid memory leaks
00070     virtual ~ULAction() = default;
00071     //cannot copy a ULAction
00072     ULAction(const ULAction& inAction) = delete;
00073     void operator=(const ULAction&) = delete;
00074
00075     virtual EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00076         bool& outWaitingForEvent, EEventID& outRaisedEvent) = 0;
00077 };
00078 }
00079 }
00080
00081 #endif // GDCMULACTION_H

```

13.577 gdcmULActionAA.h File Reference

#include "gdcmULAction.h"

Include dependency graph for gdcmULActionAA.h:



Classes

- class [gdcm::network::ULActionAA1](#)
- class [gdcm::network::ULActionAA2](#)

- class [gdcmm::network::ULActionAA3](#)
- class [gdcmm::network::ULActionAA4](#)
- class [gdcmm::network::ULActionAA5](#)
- class [gdcmm::network::ULActionAA6](#)
- class [gdcmm::network::ULActionAA7](#)
- class [gdcmm::network::ULActionAA8](#)

Namespaces

- namespace [gdcmm](#)
- namespace [gdcmm::network](#)

13.578 gdcmmULActionAA.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  *
00004  *   Copyright NumFOCUS
00005  *
00006  *   Licensed under the Apache License, Version 2.0 (the "License");
00007  *   you may not use this file except in compliance with the License.
00008  *   You may obtain a copy of the License at
00009  *
00010  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00011  *
00012  *   Unless required by applicable law or agreed to in writing, software
00013  *   distributed under the License is distributed on an "AS IS" BASIS,
00014  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00015  *   See the License for the specific language governing permissions and
00016  *   limitations under the License.
00017  *=====*/
00018  #ifndef GDCMULACTIONAA_H
00019  #define GDCMULACTIONAA_H
00020
00021  #include "gdcmmULAction.h"
00022
00023
00024  namespace gdcmm {
00025  namespace network {
00026
00027      //Send A-ABORT PDU (service-user source) and start (or restart if already started) ARTIM timer
00028      //Next State: eSta13AwaitingClose
00029      class ULActionAA1 : public ULAction {
00030      public:
00031          EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00032                                bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00033      };
00034
00035      //Stop ARTIM timer if running. Close transport connection.
00036      //Next State: eSta1Idle
00037      class ULActionAA2 : public ULAction {
00038      public:
00039          EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00040                                bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00041      };
00042
00043      //If (service-user initiated abort)
00044      //- issue A-ABORT indication and close transport connection
00045      //otherwise (service-provider initiated abort):
00046      //- issue A-P-ABORT indication and close transport connection
00047      //Next State: eSta1Idle
00048      class ULActionAA3 : public ULAction {
00049      public:
00050          EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,

```



```

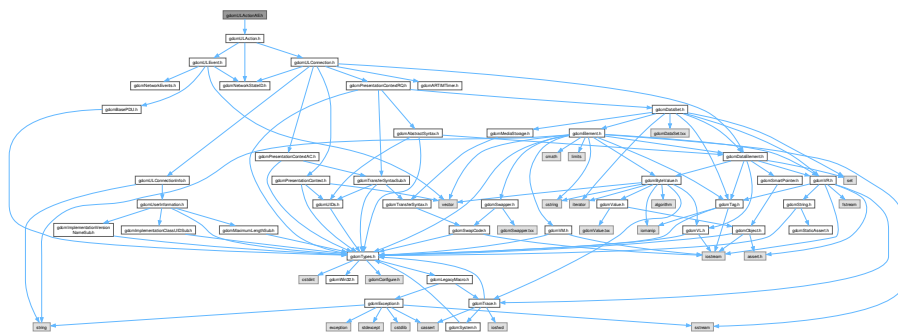
00058     bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00059 };
00060
00061 //Issue A-P-ABORT indication primitive
00062 //Next State: eSta1Idle
00063 class UActionAA4 : public UAction {
00064 public:
00065     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00066         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00067 };
00068
00069 //Stop ARTIM timer
00070 //Next State: eSta1Idle
00071 class UActionAA5 : public UAction {
00072 public:
00073     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00074         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00075 };
00076
00077 //Ignore PDU
00078 //Next State: eSta13AwaitingClose
00079 class UActionAA6 : public UAction {
00080 public:
00081     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00082         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00083 };
00084
00085 //Send A-ABORT PDU
00086 //Next State: eSta13AwaitingClose
00087 class UActionAA7 : public UAction {
00088 public:
00089     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00090         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00091 };
00092
00093 //Send A-ABORT PDU (service-provider source), issue an A-P-ABORT indication, and start ARTIM timer
00094 //Next State: eSta13AwaitingClose
00095 class UActionAA8 : public UAction {
00096 public:
00097     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00098         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00099 };
00100 }
00101 }
00102
00103 #endif // GDCMULACTIONAA_H

```

13.579 gdcMULActionAE.h File Reference

#include "gdcMULAction.h"

Include dependency graph for gdcMULActionAE.h:



Classes

- class [gdcmm::network::ULActionAE1](#)
- class [gdcmm::network::ULActionAE2](#)
- class [gdcmm::network::ULActionAE3](#)
- class [gdcmm::network::ULActionAE4](#)
- class [gdcmm::network::ULActionAE5](#)
- class [gdcmm::network::ULActionAE6](#)
- class [gdcmm::network::ULActionAE7](#)
- class [gdcmm::network::ULActionAE8](#)

Namespaces

- namespace [gdcmm](#)
- namespace [gdcmm::network](#)

13.580 gdcmmULActionAE.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMULACTIONAE_H
00019  #define GDCMULACTIONAE_H
00020
00021  #include "gdcmmULAction.h"
00022
00031
00032  namespace gdcmm {
00033  namespace network {
00034
00035      //Issue TRANSPORT CONNECT request primitive to local transport service.
00036      class ULActionAE1 : public ULAction {
00037      public:
00038          EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00039              bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00040      };
00041
00042      //Send A-ASSOCIATE-RQ-PDU
00043      //Next State: eSta5WaitRemoteAssoc
00044      class ULActionAE2 : public ULAction {
00045      public:
00046          EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00047              bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00048      };
00049
00050      //Issue A-ASSOCIATE confirmation (accept) primitive
00051      //Next State: eSta6TransferReady
00052      class ULActionAE3 : public ULAction {

```

```

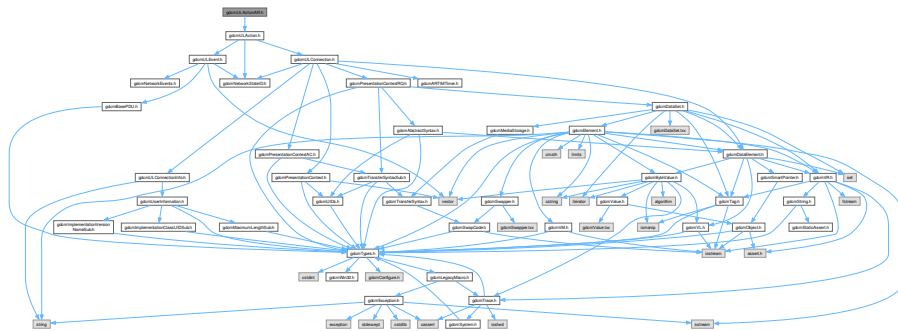
00053 public:
00054     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00055         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00056 };
00057
00058 //Issue A-ASSOCIATE confirmation (reject) primitive and close transport connection
00059 //Next State: eSta1Idle
00060 class ULActionAE4 : public ULAction {
00061 public:
00062     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00063         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00064 };
00065
00066 //Issue Transport connection response primitive, start ARTIM timer
00067 //Next State: eSta2Open
00068 class ULActionAE5 : public ULAction {
00069 public:
00070     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00071         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00072 };
00073
00074 //Stop ARTIM timer and if A-ASSOCIATE-RQ acceptable by service-provider:
00075 //- issue A-ASSOCIATE indication primitive
00076 //Next state: eSta3WaitLocalAssoc
00077 //otherwise:
00078 //- issue A-ASSOCIATE-RJ-PDU and start ARTIM timer
00079 //Next state: eSta13AwaitingClose
00080 class ULActionAE6 : public ULAction {
00081 public:
00082     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00083         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00084 };
00085
00086 //Send A-ASSOCIATE-AC PDU
00087 //Next State: eSta6TransferReady
00088 class ULActionAE7 : public ULAction {
00089 public:
00090     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00091         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00092 };
00093
00094 //Send A-ASSOCIATE-RJ PDU and start ARTIM timer
00095 //Next State: eSta13AwaitingClose
00096 class ULActionAE8 : public ULAction {
00097 public:
00098     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00099         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00100 };
00101 }
00102 }
00103 #endif // GDCMULACTIONAE_H

```

13.581 gdcMULActionAR.h File Reference

#include "gdcMULAction.h"

Include dependency graph for gdcMULActionAR.h:



Classes

- class [gdcmm::network::ULActionAR1](#)
- class [gdcmm::network::ULActionAR10](#)
- class [gdcmm::network::ULActionAR2](#)
- class [gdcmm::network::ULActionAR3](#)
- class [gdcmm::network::ULActionAR4](#)
- class [gdcmm::network::ULActionAR5](#)
- class [gdcmm::network::ULActionAR6](#)
- class [gdcmm::network::ULActionAR7](#)
- class [gdcmm::network::ULActionAR8](#)
- class [gdcmm::network::ULActionAR9](#)

Namespaces

- namespace [gdcmm](#)
- namespace [gdcmm::network](#)

13.582 gdcmmULActionAR.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  *
00004  *   Copyright NumFOCUS
00005  *
00006  *   Licensed under the Apache License, Version 2.0 (the "License");
00007  *   you may not use this file except in compliance with the License.
00008  *   You may obtain a copy of the License at
00009  *
00010  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00011  *
00012  *   Unless required by applicable law or agreed to in writing, software
00013  *   distributed under the License is distributed on an "AS IS" BASIS,
00014  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00015  *   See the License for the specific language governing permissions and
00016  *   limitations under the License.
00017  *=====*/
00018 #ifndef GDCMULACTIONAR_H
00019 #define GDCMULACTIONAR_H
00020
00021 #include "gdcmmULAction.h"
00022
00023
00024 namespace gdcmm {
00025     namespace network {
00026
00027         //Send A-RELEASE-RQ-PDU
00028         //Next State: eSta7WaitRelease
00029         class ULActionAR1 : public ULAction {
00030         public:
00031             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00032                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00033         };
00034
00035         //Issue A-RELEASE indication primitive
00036         //Next State: eSta8WaitLocalRelease
00037         class ULActionAR2 : public ULAction {
00038         public:
00039             EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00040                 bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00041         };
00042     }
00043 }

```

```

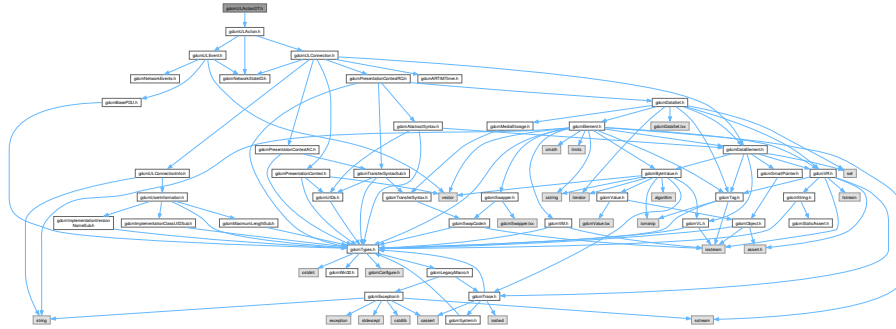
00049
00050 //Issue A-RELEASE confirmation primitive, and close transport connection
00051 //Next State: eSta1Idle
00052 class UActionAR3 : public UAction {
00053 public:
00054     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00055         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00056 };
00057
00058 //Issue A-RELEASE-RP PDU and start ARTIM timer
00059 //Next State: eSta13AwaitingClose
00060 class UActionAR4 : public UAction {
00061 public:
00062     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00063         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00064 };
00065
00066 //Stop ARTIM timer
00067 //Next State: eSta1Idle
00068 class UActionAR5 : public UAction {
00069 public:
00070     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00071         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00072 };
00073
00074 //Issue P-Data indication
00075 //Next State: eSta7WaitRelease
00076 class UActionAR6 : public UAction {
00077 public:
00078     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00079         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00080 };
00081
00082 //Issue P-DATA-TF PDU
00083 //Next State: eSta8WaitLocalRelease
00084 class UActionAR7 : public UAction {
00085 public:
00086     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00087         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00088 };
00089
00090 //Issue A-RELEASE indication (release collision):
00091 //- If association-requestor, next state is eSta9ReleaseCollisionRqLocal
00092 //- if not, next state is eSta10ReleaseCollisionAc
00093 class UActionAR8 : public UAction {
00094 public:
00095     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00096         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00097 };
00098
00099 //Send A-RELEASE-RP PDU
00100 //Next State: eSta11ReleaseCollisionRq
00101 class UActionAR9 : public UAction {
00102 public:
00103     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00104         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00105 };
00106
00107 //Issue A-RELEASE confirmation primitive
00108 //Next State: eSta12ReleaseCollisionAcLocal
00109 class UActionAR10 : public UAction {
00110 public:
00111     EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00112         bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00113 };
00114 }
00115 }
00116 #endif // GDCMULACTIONAR_H

```

13.583 gdcmULActionDT.h File Reference

#include "gdcmULAction.h"

Include dependency graph for gdcmULActionDT.h:



Classes

- class `gdcm::network::ULActionDT1`
- class `gdcm::network::ULActionDT2`

Namespaces

- namespace `gdcm`
- namespace `gdcm::network`

13.584 gdcmULActionDT.h

[Go to the documentation of this file.](#)

```
00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULACTIONDT_H
00019 #define GDCMULACTIONDT_H
00020
00021 #include "gdcmULAction.h"
00022
00030
00031 namespace gdcm {
```

```

00032 namespace network {
00033
00034     //Send P-DATA-TF PDU
00035     //Next state: eSta6TransferReady
00036     class ULActionDT1 : public ULAction {
00037     public:
00038         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00039             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00040     };
00041
00042     //Send P-DATA indication primitive
00043     //Next state: eSta6TransferReady
00044     class ULActionDT2 : public ULAction {
00045     public:
00046         EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
00047             bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
00048     };
00049 }
00050 }
00051 #endif // GDCMULACTIONDT_H

```

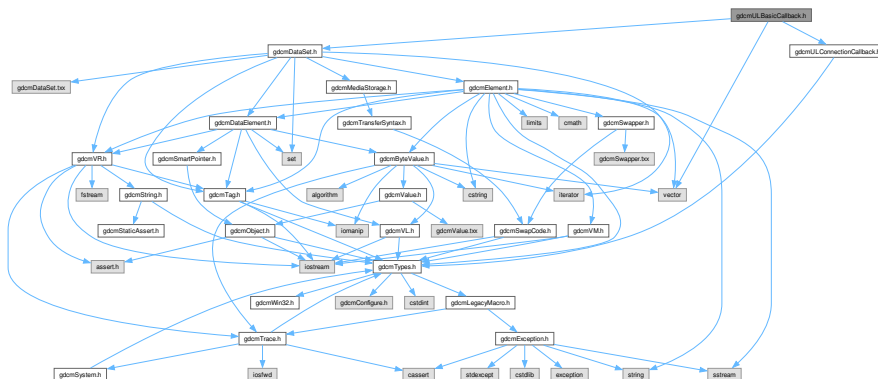
13.585 gdcmlBasicCallback.h File Reference

```

#include "gdcmlConnectionCallback.h"
#include "gdcmlDataSet.h"
#include <vector>

```

Include dependency graph for gdcmlBasicCallback.h:



Classes

- class [gdcml::network::ULBasicCallback](#)
[ULBasicCallback](#).

Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

13.586 gdcmULBasicCallback.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONBASICCALLBACK_H
00019 #define GDCMULCONNECTIONBASICCALLBACK_H
00020
00021 #include "gdcmULConnectionCallback.h"
00022 #include "gdcmDataSet.h"
00023 #include <vector>
00024
00025 namespace gdcm
00026 {
00027     namespace network
00028     {
00029         class GDCM_EXPORT ULBasicCallback : public ULConnectionCallback
00030         {
00031         {
00040             std::vector<DataSet> mDataSets;
00041             std::vector<DataSet> mResponses;
00042         public:
00043             ULBasicCallback() = default;
00044             ~ULBasicCallback() override = default; //empty, for later inheritance
00045
00046             void HandleDataSet(const DataSet& inDataSet) override;
00047             void HandleResponse(const DataSet& inDataSet) override;
00048
00049             std::vector<DataSet> const & GetDataSets() const;
00050             std::vector<DataSet> const & GetResponses() const;
00051         };
00052     } // end namespace network
00053 } // end namespace gdcm
00054
00055 #endif // GDCMULCONNECTIONBASICCALLBACK_H

```

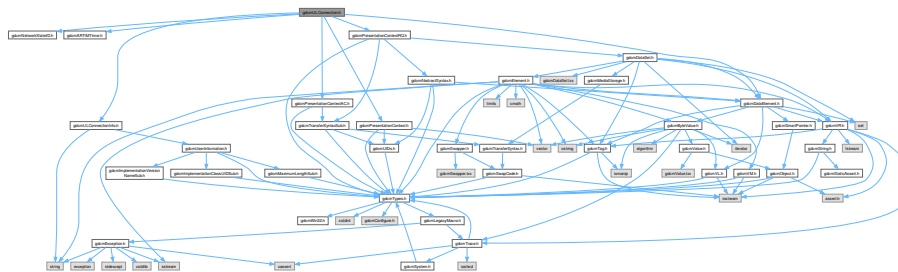
13.587 gdcmULConnection.h File Reference

```

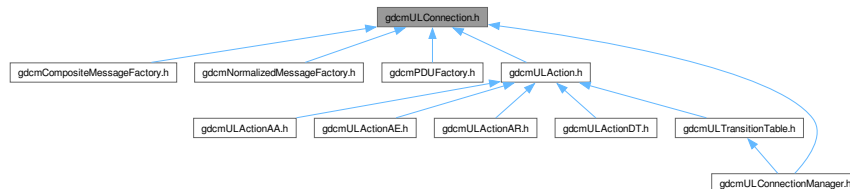
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"

```


Include dependency graph for gdcmlULConnection.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::network::ULConnection](#)
[ULConnection](#).

Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

13.588 gdcmlULConnection.h

[Go to the documentation of this file.](#)

```
00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```

00014 * See the License for the specific language governing permissions and
00015 * limitations under the License.
00016 *
00017 *=====*/
00018 #ifndef GDCMULCONNECTION_H
00019 #define GDCMULCONNECTION_H
00020
00021 #include "gdcmNetworkStateID.h"
00022 #include "gdcmARTIMTimer.h"
00023 #include "gdcmULConnectionInfo.h"
00024 #include "gdcmPresentationContextRQ.h"
00025 #include "gdcmDataElement.h"
00026 #include "gdcmPresentationContextAC.h"
00027 #include "gdcmPresentationContext.h"
00028
00029 class iosocket;
00030 class echo;
00031 namespace gdcm{
00032     namespace network{
00033
00057 class GDCM_EXPORT ULConnection
00058 {
00059     ULConnectionInfo mInfo;
00060     //this is a dirty dirty hack
00061     //but to establish an outgoing connection (scu), we need the echo service
00062     //to establish incoming, we just need a port and localhost, so an iosocket works while an
00063     //echo would fail (probably because one already exists)
00064     echo* mEcho;
00065     iosocket* mSocket; //of the three protocols offered by socket+--- echo, smtp, and ftp--
00066     //echo most closely matches what the DICOM standard describes as a network connection
00067     ARTIMTimer mTimer;
00068
00069     EStateID mCurrentState;
00070
00071     std::vector<PresentationContextRQ> mPresentationContexts;
00072     //this is our list of presentation contexts of what we can send
00073     uint32_t mMaxPDUSize;
00074
00075     std::vector<PresentationContextAC> mAcceptedPresentationContexts; //these come back from the server
00076     //and tell us what can be sent over this connection
00077
00078     TransferSyntaxSub cstorets;
00079
00080     friend class ULActionAE6;
00081     void SetCStoreTransferSyntax( TransferSyntaxSub const & ts );
00082     friend class ULConnectionManager;
00083     TransferSyntaxSub const & GetCStoreTransferSyntax( ) const;
00084 public:
00085     ULConnection(const ULConnectionInfo& inUserInformation);
00086     //destructors are virtual to prevent memory leaks by inherited classes
00087     virtual ~ULConnection();
00088
00089     EStateID GetState() const;
00090     void SetState(const EStateID& inState); //must be able to update state...
00091
00092     //echo* GetProtocol();
00093     std::iostream* GetProtocol();
00094     void StopProtocol();
00095
00096     ARTIMTimer& GetTimer();
00097
00098     const ULConnectionInfo &GetConnectionInfo() const;
00099
00100     //when the connection is first associated, the connection is told
00101     //the max packet/PDU size and the way in which to present data
00102     //(presentation contexts, etc). Store that here.
00103     void SetMaxPDUSize(uint32_t inSize);
00104     uint32_t GetMaxPDUSize() const;
00105
00106     const PresentationContextAC *GetPresentationContextACByID(uint8_t id) const;
00107     const PresentationContextRQ *GetPresentationContextRQByID(uint8_t id) const;
00108
00109     uint8_t GetPresentationContextIDFromPresentationContext(PresentationContextRQ const & pc) const;
00110
00111     std::vector<PresentationContextRQ> const & GetPresentationContexts() const;
00112     void SetPresentationContexts(const std::vector<PresentationContextRQ>& inContexts);
00113
00114     void SetPresentationContexts(const std::vector<PresentationContext>& inContexts);
00115
00116
00117

```

```

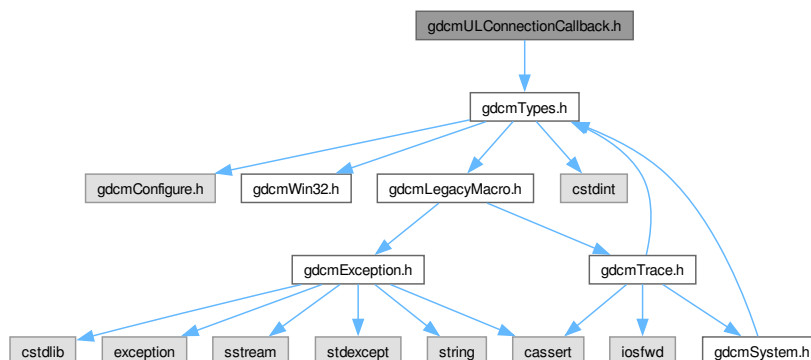
00118 //given a particular data element, presumably the SOP class,
00119 //find the presentation context for that SOP
00120 //NOT YET IMPLEMENTED
00121 PresentationContextRQ FindContext(const DataElement& de) const;
00122
00123 std::vector<PresentationContextAC> const & GetAcceptedPresentationContexts() const;
00124 std::vector<PresentationContextAC> & GetAcceptedPresentationContexts();
00125 void AddAcceptedPresentationContext(const PresentationContextAC& inPC);
00126
00128 bool InitializeConnection();
00129
00131 bool InitializeIncomingConnection();
00132
00133 ULConnection(const ULConnection&) = delete;
00134 void operator=(const ULConnection&) = delete;
00135 };
00136 }
00137 }
00138
00139 #endif // ULCONNECTION_H

```

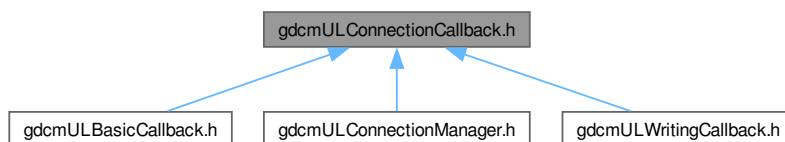
13.589 gdcmULConnectionCallback.h File Reference

#include "gdcmTypes.h"

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::network::ULConnectionCallback](#)

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.590 gdcmULConnectionCallback.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *      http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMULCONNECTIONCALLBACK_H
00019  #define GDCMULCONNECTIONCALLBACK_H
00020
00021  #include "gdcmTypes.h" //to be able to export the class
00022
00023  namespace gdcm
00024  {
00025      class DataSet;
00026      namespace network
00027      {
00028          class GDCM_EXPORT ULConnectionCallback {
00029              bool mHandledDataSet;
00030          protected:
00031              bool mImplicit;
00032              //inherited callbacks MUST call this function for the cmove loop to work properly
00033              void DataSetHandled() { mHandledDataSet = true; }
00034          public:
00035              ULConnectionCallback():mHandledDataSet(false),mImplicit(true){}
00036              virtual ~ULConnectionCallback() = default; //placeholder for inherited objects
00037              virtual void HandleDataSet(const DataSet& inDataSet) = 0;
00038              virtual void HandleResponse(const DataSet& inDataSet) = 0;
00039
00040              bool DataSetHandles() const { return mHandledDataSet; }
00041              void ResetHandledDataSet() { mHandledDataSet = false; }
00042
00043              void SetImplicitFlag( const bool imp ) { mImplicit = imp; }
00044          };
00045      }
00046  }
00047  #endif //GDCMULCONNECTIONCALLBACK_H

```

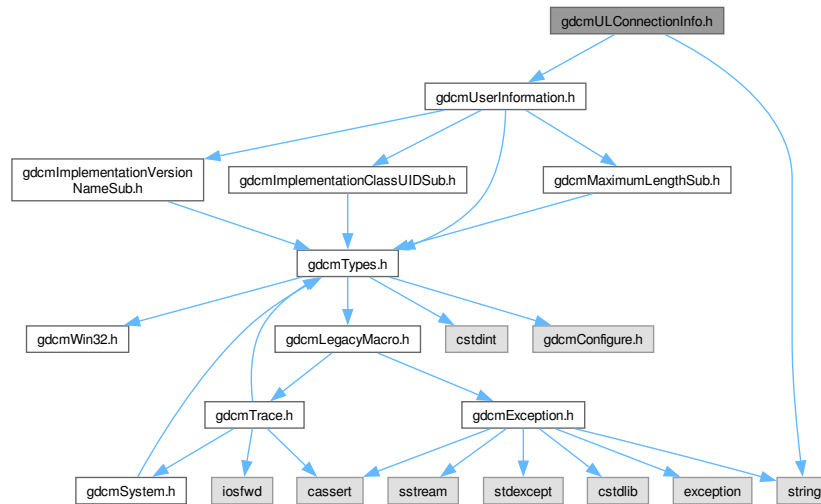
13.591 gdcmULConnectionInfo.h File Reference

```

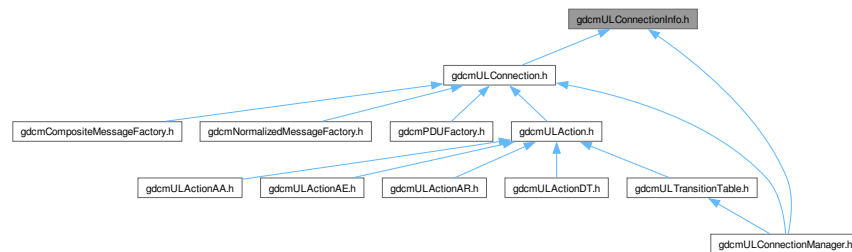
#include "gdcmUserInformation.h"
#include <string>

```

Include dependency graph for gdcmlULConnectionInfo.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::network::ULConnectionInfo](#)
[ULConnectionInfo](#).

Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

13.592 gdcmULConnectionInfo.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONINFO_H
00019 #define GDCMULCONNECTIONINFO_H
00020
00021 #include "gdcmUserInformation.h"
00022 #include <string>
00023
00024 namespace gdcm{
00025     namespace network {
00026         class ULConnectionInfo {
00027             UserInformation mUserInformation;
00028
00029             std::string mCalledAETitle;
00030             std::string mCallingAETitle;
00031
00032             unsigned long mCalledIPAddress;
00033             int mCalledIPPort;
00034             std::string mCalledComputerName; //either the IP or the name has to be filled in
00035
00036             unsigned long mMaxPDULength;
00037         public:
00038             ULConnectionInfo();
00039
00040             //it is possible to misinitialize this object, so
00041             //have it return false if something breaks (ie, given AEs are bigger than 16 characters,
00042             //no name or IP address).
00043             bool Initialize(UserInformation const &inUserInformation,
00044                 const char *inCalledAETitle, const char *inCallingAETitle,
00045                 unsigned long inCalledIPAddress, int inCalledIPPort,
00046                 std::string inCalledComputerName);
00047
00048             //UserInformation GetUserInformation() const;
00049             const char* GetCalledAETitle() const;
00050             const char* GetCallingAETitle() const;
00051
00052             unsigned long GetCalledIPAddress() const;
00053             int GetCalledIPPort() const;
00054             std::string GetCalledComputerName() const;
00055
00056             //CStore needs to know the max pdu length, so the value gets initialized
00057             //when a cstore connection is established (but not for the others).
00058             void SetMaxPDULength(unsigned long inMaxPDULength);
00059             unsigned long GetMaxPDULength() const;
00060         };
00061     }
00062 }
00063 #endif //GDCMULCONNECTIONINFO_H

```

13.593 gdcmULConnectionManager.h File Reference

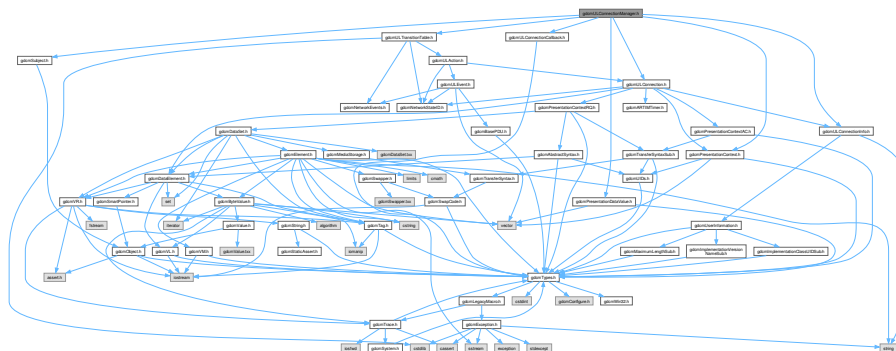
```

#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"

```

```
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmULConnectionManager.h:



Classes

- class [gdcm::network::ULConnectionManager](#)
[ULConnectionManager](#).

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.594 gdcmULConnectionManager.h

[Go to the documentation of this file.](#)

```
00001
00002 /*=====
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018 #ifndef GDCMULCONNECTIONMANAGER_H
00019 #define GDCMULCONNECTIONMANAGER_H
00020
00021 #include "gdcmULTransitionTable.h"
```

```

00022 #include "gdcmULConnection.h"
00023 #include "gdcmULConnectionInfo.h"
00024 #include "gdcmPresentationDataValue.h"
00025 #include "gdcmULConnectionCallback.h"
00026 #include "gdcmSubject.h"
00027 #include "gdcmPresentationContext.h"
00028
00029 namespace gdcm {
00030     class File;
00031     class BaseRootQuery;
00032     class BaseQuery;
00033
00034     namespace network {
00035
00045     class GDCM_EXPORT ULConnectionManager : public Subject
00046     {
00047     protected:
00048         ULConnection* mConnection;
00049         ULConnection* mSecondaryConnection;
00050         ULTransitionTable mTransitions;
00051
00052         //no copying
00053         ULConnectionManager(const ULConnectionManager& inCM);
00054
00055         //event handler loop.
00056         //will just keep running until the current event is nonexistent.
00057         //at which point, it will return the current state of the connection
00058         //this starts by initiating an action, but can be put into a passive mode
00059         //for a cmove/cstore combination by setting startWaiting to true
00060         EStateID RunEventLoop(ULEvent& inEvent, ULConnection* inWhichConnection,
00061             ULConnectionCallback* inCallback, const bool& startWaiting);
00062
00063         //like the above, but will manage the event loop for a move event (which
00064         //is basically two simultaneous connections interwoven, one inbound and
00065         //the other outbound. Note, for instance, that cmoversp's can be sent back
00066         //during the other connection's operation.
00067         EStateID RunMoveEventLoop(ULEvent& inEvent, ULConnectionCallback* inCallback);
00068
00069     public:
00070         ULConnectionManager();
00071         ~ULConnectionManager() override;
00072
00073         // NOTE: (MM) The following two functions are difficult to use, therefore marking
00074         // them as internal for now.
00075
00076         // \internal
00084         bool EstablishConnection(const std::string& inAETitle,
00085             const std::string& inConnectAETitle,
00086             const std::string& inComputerName, long inIPAddress,
00087             uint16_t inConnectPort, double inTimeout,
00088             std::vector<PresentationContext> const & pcVector );
00089
00092         bool EstablishConnectionMove(const std::string& inAETitle,
00093             const std::string& inConnectAETitle,
00094             const std::string& inComputerName, long inIPAddress,
00095             uint16_t inConnectPort, double inTimeout,
00096             uint16_t inReturnPort,
00097             std::vector<PresentationContext> const & pcVector);
00098         // \endinternal
00099
00100
00101         //bool ReestablishConnection(const EConnectionType& inConnectionType,
00102         //    const DataSet& inDS);
00103
00104         //allows for a connection to be broken, but waits for an acknowledgement
00105         //of the breaking for a certain amount of time. Returns true of the
00106         //other side acknowledges the break
00107         bool BreakConnection(const double& inTimeout);
00108
00109         //severs the connection, if it's open, without waiting for any kind of response.
00110         //typically done if the program is going down.
00111         void BreakConnectionNow();
00112
00113         //This function will send a given piece of data
00114         //across the network connection. It will return true if the
00115         //sending worked, false otherwise.
00116         //note that sending is asynchronous; as such, there's
00117         //also a 'receive' option, but that requires a callback function.
00118         //bool SendData();
00119
00120         //send the Data PDU associated with Echo (ie, a default DataPDU)

```



```

00121 //this lets the user confirm that the connection is alive.
00122 //the user should look to cout to see the response of the echo command
00123 //returns the PresentationDataValue that was returned by the remote
00124 //host. Note that the PDV can be uninitialized, which would indicate failure.
00125 //Echo does not use a callback for results.
00126 std::vector<PresentationDataValue> SendEcho();
00127
00128 // \internal
00129 // API will change...
00130 std::vector<DataSet> SendStore(const File &file, std::istream * pStream = nullptr, std::streampos dataSetOffset = 0 );
00131 std::vector<DataSet> SendFind(const BaseRootQuery* inRootQuery);
00132 std::vector<DataSet> SendMove(const BaseRootQuery* inRootQuery);
00133
00134 std::vector<DataSet> SendNEventReport (const BaseQuery* inQuery);
00135 std::vector<DataSet> SendNGet (const BaseQuery* inQuery);
00136 std::vector<DataSet> SendNSet (const BaseQuery* inQuery);
00137 std::vector<DataSet> SendNAction (const BaseQuery* inQuery);
00138 std::vector<DataSet> SendNCreate (const BaseQuery* inQuery);
00139 std::vector<DataSet> SendNDelete (const BaseQuery* inQuery);
00140 // \endinternal
00141
00143 void SendStore(const File &file, ULConnectionCallback* inCallback, std::istream * pStream = nullptr , std::streampos
dataSetOffset = 0 );
00144 void SendFind(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
00146 bool SendMove(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
00147
00148 void SendNEventReport (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00149 void SendNGet (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00150 void SendNSet (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00151 void SendNAction (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00152 void SendNCreate (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00153 void SendNDelete (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
00154
00155 };
00156 }
00157 }
00158
00159 #endif // GDCMULCONNECTIONMANAGER_H

```

13.595 gdcmlEvent.h File Reference

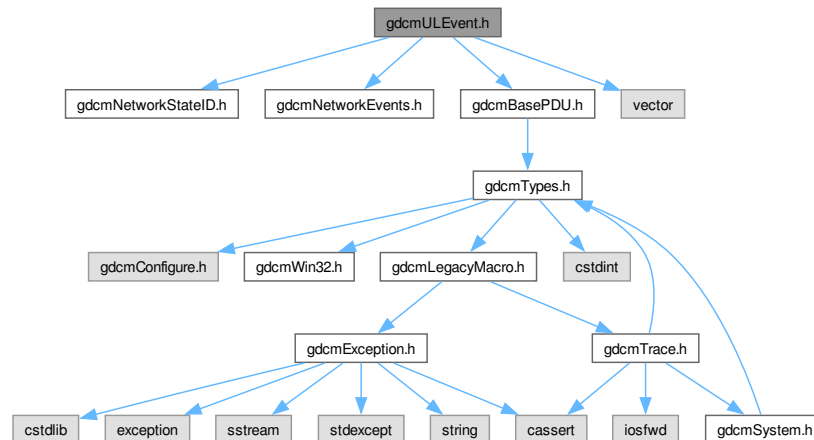
```
#include "gdcmlNetworkStateID.h"
```

```
#include "gdcmlNetworkEvents.h"
```

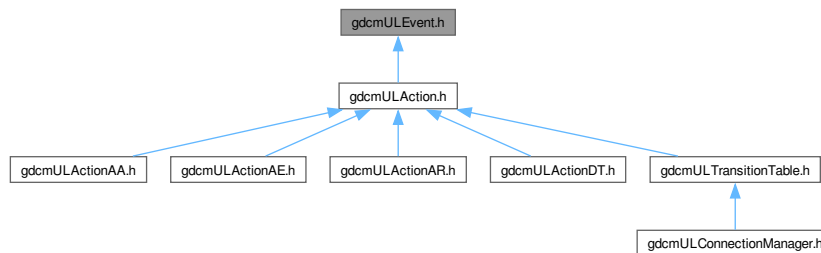
```
#include "gdcmlBasePDU.h"
```

```
#include <vector>
```

Include dependency graph for gdcmlEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcml::network::UEvent`
`UEvent`.

Namespaces

- namespace `gdcml`
- namespace `gdcml::network`

13.596 gdcmlEvent.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  * http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  *
00017  *=====*/
00018  #ifndef GDCMLEVENT_H
00019  #define GDCMLEVENT_H
00020
00021  #include "gdcmlNetworkStateID.h"
00022  #include "gdcmlNetworkEvents.h"
00023  #include "gdcmlBasePDU.h"
00024  #include <vector>
00025
00026  namespace gdcml {
00027  namespace network {
00028
00037  class UEvent {

```

```

00038     EEventID mEvent;
00039     std::vector<BasePDU*> mBasePDU;
00040     std::istream * m_pStream ;
00041     std::streampos m_posDataSet ;
00042     void DeletePDUVector(){
00043         std::vector<BasePDU*>::iterator baseItr;
00044         for (baseItr = mBasePDU.begin(); baseItr < mBasePDU.end(); baseItr++){
00045             if (*baseItr != nullptr){
00046                 delete *baseItr;
00047                 *baseItr = nullptr;
00048             }
00049         }
00050     }
00051
00052 public:
00053     ULEvent(const EEventID& inEventID, std::vector<BasePDU*> inBasePDU, std::istream * iStream = nullptr,
std::streampos posDataSet = 0 ){
00054         mEvent = inEventID;
00055         mBasePDU = inBasePDU;
00056         m_pStream = iStream ;
00057         m_posDataSet = posDataSet ;
00058     }
00059     ULEvent(const EEventID& inEventID, BasePDU* inBasePDU, std::istream * iStream = nullptr, std::streampos posDataSet
= 0 ){
00060         mEvent = inEventID;
00061         mBasePDU.push_back(inBasePDU);
00062         m_pStream = iStream ;
00063         m_posDataSet = posDataSet ;
00064     }
00065     ~ULEvent(){
00066         DeletePDUVector();
00067     }
00068
00069     EEventID GetEvent() const { return mEvent; }
00070     std::vector<BasePDU*> const & GetPDUs() const { return mBasePDU; }
00071     std::istream * GetIStream() const { return m_pStream; }
00072     std::streampos GetDataSetPos() const { return m_posDataSet; }
00073
00074     void SetEvent(const EEventID& inEvent) { mEvent = inEvent; }
00075     void SetPDU(std::vector<BasePDU*> const & inPDU) {
00076         DeletePDUVector();
00077         mBasePDU = inPDU;
00078     }
00079 };
00080 }
00081 }
00082
00083 #endif //GDCMULEVENT_H

```

13.597 gdcmlTransitionTable.h File Reference

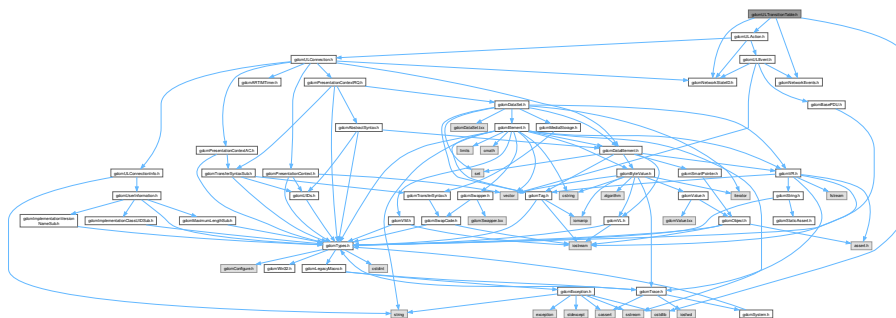
#include "gdcmlNetworkStateID.h"

#include "gdcmlNetworkEvents.h"

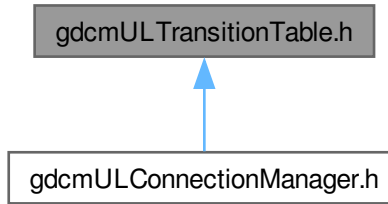
#include "gdcmlULAction.h"

#include <cstdlib>

Include dependency graph for gdcmlTransitionTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::network::TableRow](#)
- struct [gdcml::network::Transition](#)
- class [gdcml::network::ULTransitionTable](#)
[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

13.598 gdcmlTransitionTable.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  *
00004  *   Copyright NumFOCUS
00005  *
00006  *   Licensed under the Apache License, Version 2.0 (the "License");
00007  *   you may not use this file except in compliance with the License.
00008  *   You may obtain a copy of the License at
00009  *
00010  *       http://www.apache.org/licenses/LICENSE-2.0.txt
00011  *
00012  *   Unless required by applicable law or agreed to in writing, software
00013  *   distributed under the License is distributed on an "AS IS" BASIS,
00014  *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00015  *   See the License for the specific language governing permissions and
00016  *   limitations under the License.
00017  *=====*/
00018 #ifndef GDCMULTRANSITIONTABLE_H
00019 #define GDCMULTRANSITIONTABLE_H
00020
00021 #include "gdcmlNetworkStateID.h"
00022 #include "gdcmlNetworkEvents.h"
00023 #include "gdcmlULAction.h"
  
```

```

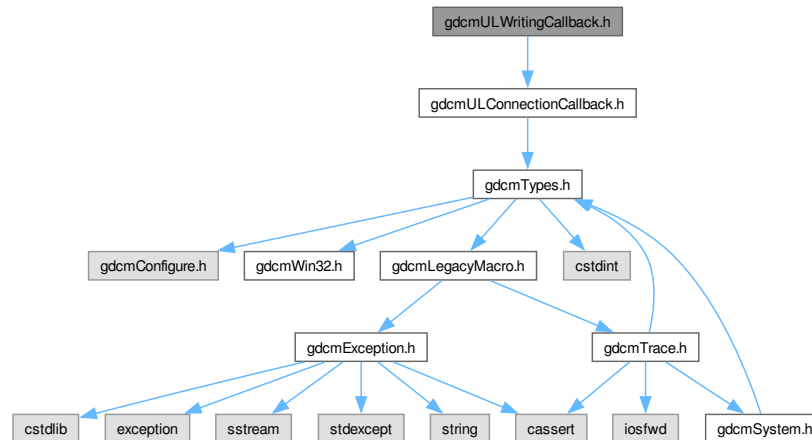
00024
00025 #include <cstdlib> // NULL
00026
00027 namespace gdcm {
00028 class Subject;
00029 namespace network{
00030 class ULConnection;
00031 class ULAction;
00032 class ULEvent;
00033
00034 //The transition dictates the action that should be taken from the start state to the end state
00035 struct Transition {
00036     int mEnd;
00037     ULAction* mAction;
00038     Transition(){
00039         mEnd = eStaDoesNotExist;
00040         mAction = nullptr;
00041     }
00042     ~Transition(){
00043         if (mAction != nullptr){
00044             delete mAction;
00045             mAction = nullptr;
00046         }
00047     }
00048     Transition(int inEndState, ULAction* inAction){
00049         mEnd = inEndState;
00050         mAction = inAction;
00051     }
00052     static Transition* MakeNew(int inEndState, ULAction* inAction){
00053         return new Transition(inEndState, inAction);
00054     }
00055 };
00056
00057 //used to define a row in table 9-10 of 3.8 2009
00058 //the transition table is events, then state,
00059 //then the transition itself (which has the event
00060 //and start state implied by their starting locations)
00061 //don't need to store the event; that's implicitly defined in the Table itself by location
00062 class TableRow{
00063 public:
00064     TableRow() {
00065         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
00066         {
00067             transitions[stateIndex] = nullptr;
00068         }
00069     }
00070     ~TableRow() {
00071         for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
00072         {
00073             Transition *t = transitions[stateIndex];
00074             delete t;
00075         }
00076     }
00077     Transition *transitions[cMaxStateID];
00078
00079     //copy constructor for stl additions into the transition table below.
00080 };
00081
00098 class ULTransitionTable
00099 {
00100 private:
00101     TableRow mTable[cMaxEventID];
00102 public:
00103     ULTransitionTable();
00104
00105     void HandleEvent(Subject*s, ULEvent& inEvent, ULConnection& inConnection,
00106         bool& outWaitingForEvent, EEventID& outRaisedEvent) const;
00107
00108     void PrintTable() const; //so that the table can be printed and verified against the DICOM standard
00109 };
00110 }
00111 }
00112 #endif // GDCMULTRANSITIONTABLE_H

```

13.599 gdcmULWritingCallback.h File Reference

#include "gdcmULConnectionCallback.h"

Include dependency graph for gdcmULWritingCallback.h:



Classes

- class [gdcm::network::ULWritingCallback](#)

Namespaces

- namespace [gdcm](#)
- namespace [gdcm::network](#)

13.600 gdcmULWritingCallback.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  *
00003  * Copyright NumFOCUS
00004  *
00005  * Licensed under the Apache License, Version 2.0 (the "License");
00006  * you may not use this file except in compliance with the License.
00007  * You may obtain a copy of the License at
00008  *
00009  *     http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  * Unless required by applicable law or agreed to in writing, software
00012  * distributed under the License is distributed on an "AS IS" BASIS,
00013  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  * See the License for the specific language governing permissions and
00015  * limitations under the License.
00016  */

```

```

00017
00018  *=====*/
00019 #ifndef GDCMULCONNECTIONWRITINGCALLBACK_H
00020 #define GDCMULCONNECTIONWRITINGCALLBACK_H
00021 #include "gdcmULConnectionCallback.h"
00022 namespace gdcm
00023 {
00024     class DataSet;
00025     namespace network
00026     {
00027         /* \brief ULWritingCallback
00028          * \details This is the most basic of callbacks for how the ULConnectionManager handles
00029          * incoming datasets. DataSets are immediately written to disk as soon as they
00030          * are received. NOTE that if the incoming connection is faster than the disk
00031          * writing speed, this callback could cause some pileups!
00032          */
00033         class GDCM_EXPORT ULWritingCallback : public ULConnectionCallback
00034         {
00035         public:
00036             std::string mDirectoryName;
00037             ULWritingCallback() = default;
00038             ~ULWritingCallback() override = default; //empty, for later inheritance
00039             void SetDirectory(const std::string& inDirectoryName) { mDirectoryName = inDirectoryName; }
00040             void HandleDataSet(const DataSet& inDataSet) override;
00041             void HandleResponse(const DataSet& inDataSet) override;
00042         };
00043     } // end namespace network
00044 } // end namespace gdcm
00045 #endif //GDCMULCONNECTIONWRITINGCALLBACK_H

```

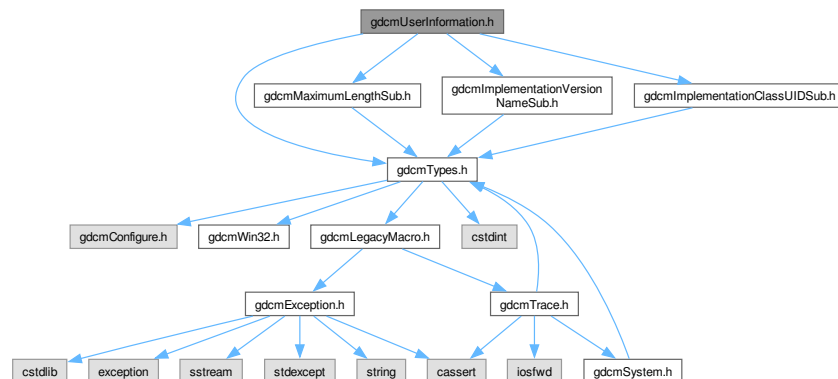
13.601 gdcmUserInformation.h File Reference

```

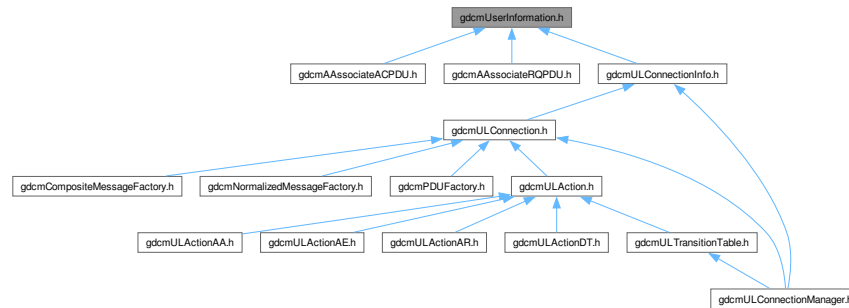
#include "gdcmTypes.h"
#include "gdcmMaximumLengthSub.h"
#include "gdcmImplementationVersionNameSub.h"
#include "gdcmImplementationClassUIDSub.h"

```

Include dependency graph for gdcmUserInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::network::UserInformation](#)
[UserInformation](#).

Namespaces

- namespace [gdcml](#)
- namespace [gdcml::network](#)

13.602 gdcmlUserInformation.h

[Go to the documentation of this file.](#)

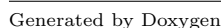
```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004 Copyright (c) 2006-2011 Mathieu Malaterre
00005 All rights reserved.
00006 See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00007
00008 This software is distributed WITHOUT ANY WARRANTY; without even
00009 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00010 PURPOSE. See the above copyright notice for more information.
00011
00012
00013 =====*/
00014 #ifndef GDCMLUSERINFORMATION_H
00015 #define GDCMLUSERINFORMATION_H
00016
00017 #include "gdcmlTypes.h"
00018 #include "gdcmlMaximumLengthSub.h"
00019 #include "gdcmlImplementationVersionNameSub.h"
00020 #include "gdcmlImplementationClassUIDSub.h"
00021
00022 namespace gdcml
00023 {
00024
00025 namespace network
00026 {
00027
00028 class AsynchronousOperationsWindowSub;

```


13.603 [gdcmWLMFindQuery.h](#) File Reference

Include dependency graph for gdcmlWLMFindQuery.h:



Classes

- class `gdcm::WLMFindQuery`
PatientRootQuery.

Namespaces

- namespace `gdcm`

13.604 `gdcmWLMFindQuery.h`

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013  #ifndef GDCMWLMFindQuery_H
00014  #define GDCMWLMFindQuery_H
00015  #include "gdcmBaseRootQuery.h"
00016
00017  namespace gdcm
00018  {
00019  class GDCM_EXPORT WLMFindQuery : public BaseRootQuery
00020  {
00021  friend class QueryFactory;
00022  public:
00023  WLMFindQuery();
00024
00025  // no sense here
00026  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00027  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00028  // validate query has required tag
00029  bool ValidateQuery(bool inStrict = true) const override;
00030
00031  UIDs::TSName GetAbstractSyntaxUID() const override;
00032  protected :
00033  DataSet GetValidDataSet() const;
00034  };
00035  } // end namespace gdcm
00036
00037  #endif // GDCMWLMFindQuery_H

```

13.605 `vtkGDCMImageReader.h` File Reference

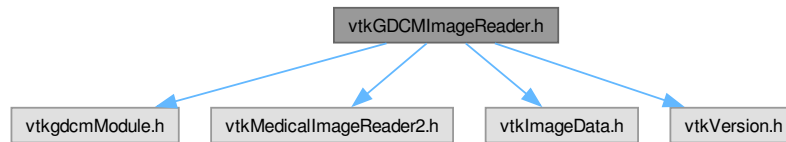
```

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"

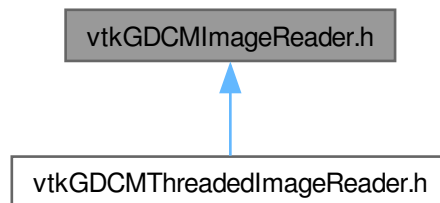
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [vtkGDCMImageReader](#)

Namespaces

- namespace [gdc](#)m

Macros

- #define [VTK_CMYK](#) 8
- #define [VTK_INVERSE_LUMINANCE](#) 5
- #define [VTK_LOOKUP_TABLE](#) 6
- #define [VTK_YBR](#) 7

13.605.1 Macro Definition Documentation

13.605.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

13.605.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

13.605.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

13.605.1.4 VTK_YBR

```
#define VTK_YBR 7
```

13.606 vtkGDCMImageReader.h

[Go to the documentation of this file.](#)

```
00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkGDCMImageReader - read DICOM Image files (Pixel Data)
00015  // .SECTION Description
00016  // vtkGDCMImageReader is a source object that reads some DICOM files
00017  // this reader is single threaded.
00018  // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00019  // upside down as VTK would expect, use this option only if you know what you are doing.
00020  // .SECTION Implementation note: when reading a series of 2D slices, user is
00021  // expected to provide an ordered list of filenames. No sorting will be applied afterward.
00022  // .SECTION Implementation note: Although 99% of the time the Zspacing as read
00023  // from a tag in a 2D DICOM file should be correct, there has been reports that this
00024  // value can be missing, or incorrect, in which case users are advised to override this
00025  // value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
00026  // vtkImageChangeInformation on the reader itself.
00027  // .SECTION TODO
00028  // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
00029  // list of 2D files are supported for now.
00030  // .SECTION TODO
00031  // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
00032  // .SECTION BUG
00033  // Overlay are assumed to have the same extent as image. Right now if overlay origin is not
00034  // 0,0 the overlay will have an offset...
00035  // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the file)
```

```

00036 // .SECTION DataOrigin
00037 // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
00038 // identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the DataOrigin
00039 // is then translated to the other side of the image.
00040 // .SECTION Spacing
00041 // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect up-side-down)
00042 // as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame object
00043 // contains image which do not represent uniform volume.
00044 // .SECTION Warning
00045 // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
00046 // it is *required* that FileLowerLeft is set to ON as coordinate system
00047 // would be inconsistent in between the two data structures.
00048 // .SECTION Color Space mapping:
00049 // * VTK_LUMINANCE <-> MONOCHROME2
00050 // * VTK_LUMINANCE_ALPHA <-> Not supported
00051 // * VTK_RGB <-> RGB
00052 // * VTK_RGBA <-> ARGB (deprecated, DICOM 2008)
00053 // * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
00054 // * VTK_LOOKUP_TABLE <-> PALETTE COLOR
00055 // * VTK_YBR <-> YBR_FULL
00056 //
00057 // For detailed information on color space transformation and true lossless transformation see:
00058 // http://gdcm.sourceforge.net/wiki/index.php/Color\_Space\_Transformations
00059
00060 // .SECTION See Also
00061 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
00062 // vtkDICOMImageReader
00063
00064 #ifndef VTKGDCMIMAGEREADER_H
00065 #define VTKGDCMIMAGEREADER_H
00066
00067 #include "vtkgdcmModule.h"
00068 #include "vtkMedicalImageReader2.h"
00069 #include "vtkImageData.h"
00070 #include "vtkVersion.h"
00071
00072 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00073 #else
00074 class vtkMedicalImageProperties;
00075 #endif
00076 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00077 #else
00078 class vtkStringArray;
00079 #endif
00080 class vtkPolyData;
00081
00082 // vtkSystemIncludes.h defines:
00083 // #define VTK_LUMINANCE 1
00084 // #define VTK_LUMINANCE_ALPHA 2
00085 // #define VTK_RGB 3
00086 // #define VTK_RGBA 4
00087 #ifndef VTK_INVERSE_LUMINANCE
00088 #define VTK_INVERSE_LUMINANCE 5
00089 #endif
00090 #ifndef VTK_LOOKUP_TABLE
00091 #define VTK_LOOKUP_TABLE 6
00092 #endif
00093 #ifndef VTK_YBR
00094 #define VTK_YBR 7
00095 #endif
00096 #ifndef VTK_CMYK
00097 #define VTK_CMYK 8
00098 #endif
00099
00100 //BTX
00101 namespace gdcm { class ImageReader; }
00102 //ETX
00103 class vtkMatrix4x4;
00104 class VTKGDCM_EXPORT vtkGDCMImageReader : public vtkMedicalImageReader2
00105 {
00106 public:
00107 static vtkGDCMImageReader *New();
00108 vtkTypeMacro(vtkGDCMImageReader,vtkMedicalImageReader2);
00109 virtual void PrintSelf(ostream& os, vtkIndent indent);
00110
00111 // Description: is the given file name a DICOM file containing an image ?
00112 virtual int CanReadFile(const char* fname);
00113
00114 // Description:
00115 // Valid extensions
00116 virtual const char* GetFileExtensions()

```

```

00117 {
00118 // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
00119 return ".dcm .DCM";
00120 }
00121
00122 // Description:
00123 // A descriptive name for this format
00124 virtual const char* GetDescriptiveName()
00125 {
00126 return "DICOM";
00127 }
00128
00129 // Description:
00130 // Get the Image Position (Patient) as stored in the DICOM file
00131 // This is a read-only data member
00132 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00133
00134 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00135 #else
00136 // Description:
00137 // Get the medical image properties object
00138 vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00139 #endif
00140 virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00141
00142 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00143 #else
00144 virtual void SetFileNames(vtkStringArray*);
00145 vtkGetObjectMacro(FileNames, vtkStringArray);
00146 #endif
00147
00148 // Description:
00149 // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
00150 // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
00151 vtkGetMacro(LoadOverlays,int);
00152 vtkSetMacro(LoadOverlays,int);
00153 vtkBooleanMacro(LoadOverlays,int);
00154
00155 // Description:
00156 // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
00157 vtkGetMacro(LoadIconImage,int);
00158 vtkSetMacro(LoadIconImage,int);
00159 vtkBooleanMacro(LoadIconImage,int);
00160
00161 // Description:
00162 // Set/Get whether or not the image was compressed using a lossy compression algorithm
00163 vtkGetMacro(LossyFlag,int);
00164 vtkSetMacro(LossyFlag,int);
00165 vtkBooleanMacro(LossyFlag,int);
00166
00167 // Description:
00168 // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
00169 // Only valid when LoadOverlays is true
00170 vtkGetMacro(NumberOfOverlays,int);
00171
00172 // Description:
00173 // Read only: number of icon image (there can only be zero or one icon per file)
00174 // Only valid when LoadIconImage is true
00175 vtkGetMacro(NumberOfIconImages,int);
00176
00177 // Description:
00178 // Get Overlay/IconImage
00179 // Remember to ALWAYS use those methods in your code, as the internal number for the output port
00180 // is not guaranteed to remain the same, as features are added to the reader
00181 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00182 //FIXME: Need to get rid of BTX/ETX if only the Python Wrapper of VTK 4.2 would let me
00183 //BTX
00184 vtkAlgorithmOutput* GetOverlayPort(int index);
00185 vtkAlgorithmOutput* GetIconImagePort();
00186 //ETX
00187 #endif
00188 vtkImageData* GetOverlay(int i);
00189 vtkImageData* GetIconImage();
00190
00191 // Description:
00192 // Load image with its associated Lookup Table
00193 vtkGetMacro(ApplyLookupTable,int);
00194 vtkSetMacro(ApplyLookupTable,int);
00195 vtkBooleanMacro(ApplyLookupTable,int);
00196
00197 // Description:

```

```

00198 // Load image as YBR
00199 vtkGetMacro(ApplyYBRToRGB,int)
00200 vtkSetMacro(ApplyYBRToRGB,int)
00201 vtkBooleanMacro(ApplyYBRToRGB,int);
00202
00203 // Description:
00204 // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE,
VTK_YBR or VTK_CMYK
00205 // or 0 when ImageFormat is not handled.
00206 // Warning: For color image, PlanarConfiguration need to be taken into account.
00207 vtkGetMacro(ImageFormat,int);
00208
00209 // Description:
00210 // Return the Planar Configuration. This simply means that the internal DICOM image was stored
00211 // using a particular planar configuration (most of the time: 0)
00212 // For monochrome image, PlanarConfiguration is always 0
00213 vtkGetMacro(PlanarConfiguration,int);
00214
00215 // Description:
00216 // Return the 'raw' information stored in the DICOM file:
00217 // In case of a series of multiple files, only the first file is considered. The Image Orientation (Patient)
00218 // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
00219 // using the ZSpacing (3rd dimension)
00220 // (0020,0032) DS [87.774866\ -182.908510\ 168.629671] # 32, 3 ImagePositionPatient
00221 // (0020,0037) DS [0.001479\ 0.999989\ -0.004376\ -0.002039\ -0.004372\ -0.999988] # 58, 6 ImageOrientationPatient
00222 vtkGetVector3Macro(ImagePositionPatient,double);
00223 vtkGetVector6Macro(ImageOrientationPatient,double);
00224
00225 // Description:
00226 // Set/Get the first Curve Data:
00227 vtkGetObjectMacro(Curve,vtkPolyData);
00228 virtual void SetCurve(vtkPolyData *pd);
00229
00230 // Description:
00231 // \DEPRECATED:
00232 // Modality LUT
00233 // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
00234 // varying along the Series read. Therefore user are advices not to use those functions
00235 // anymore
00236 vtkGetMacro(Shift,double);
00237 vtkGetMacro(Scale,double);
00238
00239 protected:
00240 vtkGDCMImageReader();
00241 ~vtkGDCMImageReader();
00242
00243 vtkSetVector6Macro(ImageOrientationPatient,double);
00244
00245 //BTX
00246 void FillMedicalImageInformation(const gdcm::ImageReader &reader);
00247 //ETX
00248 int RequestInformationCompat();
00249 int RequestDataCompat();
00250
00251 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00252 int ProcessRequest(vtkInformation* request,
00253                   vtkInformationVector** inputVector,
00254                   vtkInformationVector* outputVector);
00255 int RequestInformation(vtkInformation *request,
00256                       vtkInformationVector **inputVector,
00257                       vtkInformationVector *outputVector);
00258 int RequestData(vtkInformation *request,
00259                vtkInformationVector **inputVector,
00260                vtkInformationVector *outputVector);
00261 #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00262 void ExecuteInformation();
00263 void ExecuteData(vtkDataObject *out);
00264 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00265
00266 protected:
00267 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00268 #else
00269 // Description:
00270 // Medical Image properties
00271 vtkMedicalImageProperties *MedicalImageProperties;
00272 #endif
00273 #if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
00274 #else
00275 vtkStringArray *FileNames;
00276 #endif
00277

```

```

00278 vtkMatrix4x4 *DirectionCosines;
00279 int LoadOverlays;
00280 int NumberOfOverlays;
00281 int LoadIconImage;
00282 int NumberOfIconImages;
00283 int IconImageDataExtent[6];
00284 double ImagePositionPatient[3];
00285 double ImageOrientationPatient[6];
00286 vtkPolyData *Curve;
00287
00288 int ImageFormat;
00289 // the following 3, should remain optional
00290 int ApplyInverseVideo;
00291 int ApplyLookupTable;
00292 int ApplyYBRToRGB;
00293 // I think that planar configuration need to always be applied as far as VTK is concerned
00294 int ApplyPlanarConfiguration;
00295 int ApplyShiftScale;
00296
00297 int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
00298
00299 double Shift;
00300 double Scale;
00301 int IconDataScalarType;
00302 int IconNumberOfScalarComponents;
00303 int PlanarConfiguration;
00304 int LossyFlag;
00305 int ForceRescale;
00306
00307 protected:
00308 // TODO / FIXME
00309 void SetFilePrefix(const char *) {}
00310 vtkGetStringMacro(FilePrefix);
00311 void SetFilePattern(const char *) {}
00312 vtkGetStringMacro(FilePattern);
00313
00314 private:
00315 vtkGDCMImageReader(const vtkGDCMImageReader&); // Not implemented.
00316 void operator=(const vtkGDCMImageReader&); // Not implemented.
00317 };
00318 #endif

```

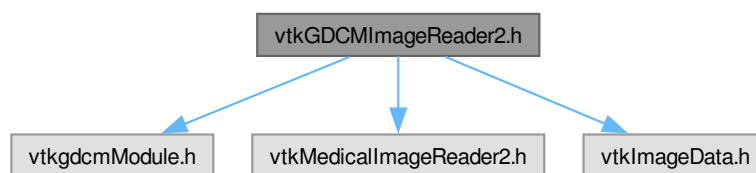
13.607 vtkGDCMImageReader2.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"

```

Include dependency graph for vtkGDCMImageReader2.h:



Classes

- class [vtkGDCMImageReader2](#)

Namespaces

- namespace [gdc](#)

Macros

- `#define VTK_CMYK` 8
- `#define VTK_INVERSE_LUMINANCE` 5
- `#define VTK_LOOKUP_TABLE` 6
- `#define VTK_YBR` 7

13.607.1 Macro Definition Documentation

13.607.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

13.607.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

13.607.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

13.607.1.4 VTK_YBR

```
#define VTK_YBR 7
```

13.608 vtkGDCMImageReader2.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkGDCMImageReader2 - read DICOM Image files (Pixel Data)
00015  // .SECTION Description
00016  // vtkGDCMImageReader2 is a source object that reads some DICOM files
00017  // this reader is single threaded.
00018  // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00019  // upside down as VTK would expect, use this option only if you know what you are doing.
00020  // .SECTION Implementation note: when reading a series of 2D slices, user is
00021  // expected to provide an ordered list of filenames. No sorting will be applied afterward.
00022  // .SECTION Implementation note: Although 99% of the time the Zspacing as read
00023  // from a tag in a 2D DICOM file should be correct, there has been reports that this
00024  // value can be missing, or incorrect, in which case users are advised to override this
00025  // value using the return value from gdcml::IPPSorter::GetZSpacing() and set it via
00026  // vtkImageChangeInformation on the reader itself.
00027  // .SECTION TODO
00028  // This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
00029  // list of 2D files are supported for now.
00030  // .SECTION TODO
00031  // Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
00032  // .SECTION BUG
00033  // Overlay are assumed to have the same extent as image. Right now if overlay origin is not
00034  // 0,0 the overlay will have an offset...
00035  // Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the file)
00036  // .SECTION DataOrigin
00037  // When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
00038  // identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the DataOrigin
00039  // is then translated to the other side of the image.
00040  // .SECTION Spacing
00041  // When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect up-side-down)
00042  // as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame object
00043  // contains image which do not represent uniform volume.
00044  // .SECTION Warning
00045  // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader2
00046  // it is *required* that FileLowerLeft is set to ON as coordinate system
00047  // would be inconsistent in between the two data structures.
00048  // .SECTION Color Space mapping:
00049  // * VTK_LUMINANCE <-> MONOCHROME2
00050  // * VTK_LUMINANCE_ALPHA <-> Not supported
00051  // * VTK_RGB <-> RGB
00052  // * VTK_RGBA <-> ARGB (deprecated, DICOM 2008)
00053  // * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
00054  // * VTK_LOOKUP_TABLE <-> PALETTE COLOR
00055  // * VTK_YBR <-> YBR_FULL
00056  //
00057  // For detailed information on color space transformation and true lossless transformation see:
00058  // http://gdcml.sourceforge.net/wiki/index.php/Color_Space_Transformations
00059
00060  // .SECTION See Also
00061  // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
00062  // vtkDICOMImageReader
00063
00064  #ifndef VTKGDCMIMAGEREADER2_H
00065  #define VTKGDCMIMAGEREADER2_H
00066
00067  #include "vtkgdcmModule.h"
00068  #include "vtkMedicalImageReader2.h"
00069  #include "vtkImageData.h"
00070
00071  class vtkPolyData;
00072
00073  // vtkSystemIncludes.h defines:

```

```

00074 // #define VTK_LUMINANCE      1
00075 // #define VTK_LUMINANCE_ALPHA 2
00076 // #define VTK_RGB             3
00077 // #define VTK_RGBA            4
00078 #ifndef VTK_INVERSE_LUMINANCE
00079 #define VTK_INVERSE_LUMINANCE 5
00080 #endif
00081 #ifndef VTK_LOOKUP_TABLE
00082 #define VTK_LOOKUP_TABLE 6
00083 #endif
00084 #ifndef VTK_YBR
00085 #define VTK_YBR 7
00086 #endif
00087 #ifndef VTK_CMYK
00088 #define VTK_CMYK 8
00089 #endif
00090
00091 //BTX
00092 namespace gdcmm { class ImageReader; }
00093 //ETX
00094 class vtkMatrix4x4;
00095 class VTKGDCM_EXPORT vtkGDCMImageReader2 : public vtkMedicalImageReader2
00096 {
00097 public:
00098     static vtkGDCMImageReader2 *New();
00099     vtkTypeMacro(vtkGDCMImageReader2,vtkMedicalImageReader2);
00100     virtual void PrintSelf(ostream& os, vtkIndent indent);
00101
00102     // Description: is the given file name a DICOM file containing an image ?
00103     virtual int CanReadFile(const char* fname);
00104
00105     // Description:
00106     // Valid extensions
00107     virtual const char* GetFileExtensions()
00108     {
00109         // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
00110         return ".dcm .DCM";
00111     }
00112
00113     // Description:
00114     // A descriptive name for this format
00115     virtual const char* GetDescriptiveName()
00116     {
00117         return "DICOM";
00118     }
00119
00120     // Description:
00121     // Get the Image Position (Patient) as stored in the DICOM file
00122     // This is a read-only data member
00123     vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00124
00125     virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00126
00127     // Description:
00128     // Specifically request to load the overlay into the gdcmm-VTK layer (gdcmm always loads them when found).
00129     // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
00130     vtkGetMacro(LoadOverlays,int);
00131     vtkSetMacro(LoadOverlays,int);
00132     vtkBooleanMacro(LoadOverlays,int);
00133
00134     // Description:
00135     // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
00136     vtkGetMacro(LoadIconImage,int);
00137     vtkSetMacro(LoadIconImage,int);
00138     vtkBooleanMacro(LoadIconImage,int);
00139
00140     // Description:
00141     // Set/Get whether or not the image was compressed using a lossy compression algorithm
00142     vtkGetMacro(LossyFlag,int);
00143     vtkSetMacro(LossyFlag,int);
00144     vtkBooleanMacro(LossyFlag,int);
00145
00146     // Description:
00147     // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
00148     // Only valid when LoadOverlays is true
00149     vtkGetMacro(NumberOfOverlays,int);
00150
00151     // Description:
00152     // Read only: number of icon image (there can only be zero or one icon per file)
00153     // Only valid when LoadIconImage is true
00154     vtkGetMacro(NumberOfIconImages,int);

```

```

00155
00156 // Description:
00157 // Get Overlay/IconImage
00158 // Remember to ALWAYS use those methods in your code, as the internal number for the output port
00159 // is not guaranteed to remain the same, as features are added to the reader
00160 vtkAlgorithmOutput* GetOverlayPort(int index);
00161 vtkAlgorithmOutput* GetIconImagePort();
00162 vtkImageData* GetOverlay(int i);
00163 vtkImageData* GetIconImage();
00164
00165 // Description:
00166 // Load image with its associated Lookup Table
00167 vtkGetMacro(ApplyLookupTable,int);
00168 vtkSetMacro(ApplyLookupTable,int);
00169 vtkBooleanMacro(ApplyLookupTable,int);
00170
00171 // Description:
00172 // Load image as YBR
00173 vtkGetMacro(ApplyYBRToRGB,int)
00174 vtkSetMacro(ApplyYBRToRGB,int)
00175 vtkBooleanMacro(ApplyYBRToRGB,int);
00176
00177 // Description:
00178 // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE,
VTK_YBR or VTK_CMYK
00179 // or 0 when ImageFormat is not handled.
00180 // Warning: For color image, PlanarConfiguration need to be taken into account.
00181 vtkGetMacro(ImageFormat,int);
00182
00183 // Description:
00184 // Return the Planar Configuration. This simply means that the internal DICOM image was stored
00185 // using a particular planar configuration (most of the time: 0)
00186 // For monochrome image, PlanarConfiguration is always 0
00187 vtkGetMacro(PlanarConfiguration,int);
00188
00189 // Description:
00190 // Return the 'raw' information stored in the DICOM file:
00191 // In case of a series of multiple files, only the first file is considered. The Image Orientation (Patient)
00192 // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
00193 // using the ZSpacing (3rd dimension)
00194 // (0020,0032) DS [87.774866\ -182.908510\ 168.629671] # 32, 3 ImagePositionPatient
00195 // (0020,0037) DS [0.001479\ 0.999989\ -0.004376\ -0.002039\ -0.004372\ -0.999988] # 58, 6 ImageOrientationPatient
00196 vtkGetVector3Macro(ImagePositionPatient,double);
00197 vtkGetVector6Macro(ImageOrientationPatient,double);
00198
00199 // Description:
00200 // Set/Get the first Curve Data:
00201 vtkGetObjectMacro(Curve,vtkPolyData);
00202 virtual void SetCurve(vtkPolyData *pd);
00203
00204 // Description:
00205 // \DEPRECATED:
00206 // Modality LUT
00207 // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
00208 // varying along the Series read. Therefore user are advices not to use those functions
00209 // anymore
00210 vtkGetMacro(Shift,double);
00211 vtkGetMacro(Scale,double);
00212
00213 protected:
00214 vtkGDCMImageReader2();
00215 ~vtkGDCMImageReader2();
00216
00217 vtkSetVector6Macro(ImageOrientationPatient,double);
00218
00219 //BTX
00220 void FillMedicalImageInformation(const gdcm::ImageReader &reader);
00221 //ETX
00222 int RequestInformationCompat();
00223 int RequestDataCompat();
00224
00225 int ProcessRequest(vtkInformation* request,
00226                   vtkInformationVector** inputVector,
00227                   vtkInformationVector* outputVector);
00228 int RequestInformation(vtkInformation *request,
00229                       vtkInformationVector **inputVector,
00230                       vtkInformationVector *outputVector);
00231 int RequestData(vtkInformation *request,
00232                 vtkInformationVector **inputVector,
00233                 vtkInformationVector *outputVector);
00234

```

```

00235 protected:
00236   vtkMatrix4x4 *DirectionCosines;
00237   int LoadOverlays;
00238   int NumberOfOverlays;
00239   int LoadIconImage;
00240   int NumberOfIconImages;
00241   int IconImageDataExtent[6];
00242   double ImagePositionPatient[3];
00243   double ImageOrientationPatient[6];
00244   vtkPolyData *Curve;
00245
00246   int ImageFormat;
00247   // the following 3, should remain optional
00248   int ApplyInverseVideo;
00249   int ApplyLookupTable;
00250   int ApplyYBRToRGB;
00251   // I think that planar configuration need to always be applied as far as VTK is concerned
00252   int ApplyPlanarConfiguration;
00253   int ApplyShiftScale;
00254
00255   int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);
00256
00257   double Shift;
00258   double Scale;
00259   int IconDataScalarType;
00260   int IconNumberOfScalarComponents;
00261   int PlanarConfiguration;
00262   int LossyFlag;
00263   int ForceRescale;
00264
00265 protected:
00266   // TODO / FIXME
00267   void SetFilePrefix(const char *) {}
00268   vtkGetStringMacro(FilePrefix);
00269   void SetFilePattern(const char *) {}
00270   vtkGetStringMacro(FilePattern);
00271
00272 private:
00273   vtkGDCMImageReader2(const vtkGDCMImageReader2&); // Not implemented.
00274   void operator=(const vtkGDCMImageReader2&); // Not implemented.
00275 };
00276 #endif

```

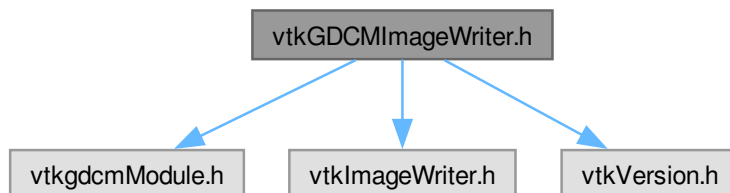
13.609 vtkGDCMImageWriter.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkImageWriter.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



Classes

- class [vtkGDCMImageWriter](#)

13.610 vtkGDCMImageWriter.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014 // .NAME vtkGDCMImageWriter - write DICOM files
00015 // .SECTION Description
00016 // vtkGDCMImageWriter is a sink object that write DICOM files
00017 // this writer is single threaded (see vtkGDCMThreadedImageReader2 for multi-thread)
00018 //
00019 // .SECTION Warning: vtkLookupTable from the vtkImageData object taken into account
00020 // only if ImageFormat is set to VTK_LOOKUP_TABLE
00021 //
00022 // .SECTION NOTE We are not using the usual API SetFilePrefix / SetFilePattern,
00023 // but instead a list of filenames: see SetFileNames and class gdcm::FilenameGenerator
00024 //
00025 // .SECTION Warning
00026 // You need to specify the correct ImageFormat (taken from the reader)
00027 // You need to explicitly specify the DirectionCosines (taken from the reader)
00028 // Since VTK 5.4 vtkMedicalImageProperties has its own DirectionCosine (no 's')
00029 // user need to make sure the vtkMatrix4x4 is compatible with the 6-vector DirectionCosine.
00030 //
00031 // .SECTION NOTE Shift/Scale are global to all DICOM frames (=files) written
00032 // as 2D slice, therefore the shift/scale operation might not be optimized for
00033 // all slices. This is not recommended for image with a large dynamic range.
00034 //
00035 // .SECTION See Also
00036 // vtkImageWriter vtkMedicalImageProperties vtkGDCMImageReader
00037
00038 #ifndef VTKGDCMIMAGEWRITER_H
00039 #define VTKGDCMIMAGEWRITER_H
00040
00041 #include "vtkgdcmModule.h"
00042 #include "vtkImageWriter.h"
00043 #include "vtkVersion.h"
00044
00045 class vtkLookupTable;
00046 class vtkMedicalImageProperties;
00047 class vtkMatrix4x4;
00048 class vtkStringArray;
00049 class VTKGDCM_EXPORT vtkGDCMImageWriter : public vtkImageWriter
00050 {
00051 public:
00052   static vtkGDCMImageWriter *New();
00053   vtkTypeMacro(vtkGDCMImageWriter,vtkImageWriter);
00054   virtual void PrintSelf(ostream& os, vtkIndent indent);
00055
00056   // Description:
00057   // Pass in the vtkmedicalimageproperties object for medical information
00058   // to be mapped to DICOM attributes.
00059   vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00060   virtual void SetMedicalImageProperties(vtkMedicalImageProperties*);
00061
00062   // Description:
00063   // Pass in the list of filename to be used to write out the DICOM file(s)
00064   virtual void SetFileNames(vtkStringArray*);
00065   vtkGetObjectMacro(FileNames, vtkStringArray);

```

```

00066
00067 // Description:
00068 // Set/Get whether or not the image was compressed using a lossy compression algorithm
00069 vtkGetMacro(LossyFlag,int);
00070 vtkSetMacro(LossyFlag,int);
00071 vtkBooleanMacro(LossyFlag,int);
00072
00073 // I need that...
00074 virtual void Write();
00075
00076 // Description:
00077 // Get the extension for this file format.
00078 virtual const char* GetFileExtensions() {
00079     return ".dcm .DCM"; }
00080
00081 // Description:
00082 // Get the name of this file format.
00083 virtual const char* GetDescriptiveName() {
00084     return "DICOM"; }
00085
00086 // Description:
00087 // You need to manually specify the direction the image is in to write a valid DICOM file
00088 // since vtkImageData do not contains one (eg. MR Image Storage, CT Image Storage...)
00089 virtual void SetDirectionCosines(vtkMatrix4x4 *matrix);
00090 vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
00091 virtual void SetDirectionCosinesFromImageOrientationPatient(const double dircos[6]);
00092
00093 // Description:
00094 // Modality LUT
00095 vtkSetMacro(Shift, double);
00096 vtkGetMacro(Shift, double);
00097 vtkSetMacro(Scale, double);
00098 vtkGetMacro(Scale, double);
00099
00100 // Description:
00101 // See vtkGDCMImageReader for list of ImageFormat
00102 vtkGetMacro(ImageFormat,int);
00103 vtkSetMacro(ImageFormat,int);
00104
00105 // Description:
00106 // Set/Get whether the data comes from the file starting in the lower left
00107 // corner or upper left corner.
00108 vtkBooleanMacro(FileLowerLeft, int);
00109 vtkGetMacro(FileLowerLeft, int);
00110 vtkSetMacro(FileLowerLeft, int);
00111
00112 // Description:
00113 // For color image (more than a single comp) you can specify the planar configuration you prefer
00114 vtkSetMacro(PlanarConfiguration,int);
00115 vtkGetMacro(PlanarConfiguration,int);
00116
00117 // Description:
00118 // Set/Get specific StudyUID / SeriesUID
00119 vtkSetStringMacro(StudyUID);
00120 vtkGetStringMacro(StudyUID);
00121 vtkSetStringMacro(SeriesUID);
00122 vtkGetStringMacro(SeriesUID);
00123
00124 //BTX
00125 enum CompressionTypes {
00126     NO_COMPRESSION = 0, // raw (default)
00127     JPEG_COMPRESSION, // JPEG
00128     JPEG2000_COMPRESSION, // J2K
00129     JPEGLS_COMPRESSION, // JPEG-LS
00130     RLE_COMPRESSION // RLE
00131 };
00132 //ETX
00133 // Set/Get the compression type
00134 vtkSetMacro(CompressionType, int);
00135 vtkGetMacro(CompressionType, int);
00136
00137 //void SetCompressionTypeFromString(const char *);
00138 //const char *GetCompressionTypeAsString();
00139
00140 protected:
00141     vtkGDCMImageWriter();
00142     ~vtkGDCMImageWriter();
00143
00144 #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00145     int FillInputPortInformation(int port, vtkInformation *info);
00146     int RequestInformation(

```

```

00147     vtkInformation *request,
00148     vtkInformationVector **inputVector,
00149     vtkInformationVector *outputVector);
00150 int RequestUpdateExtent(
00151     vtkInformation *request,
00152     vtkInformationVector **inputVector,
00153     vtkInformationVector *outputVector);
00154 int RequestData(
00155     vtkInformation *request,
00156     vtkInformationVector **inputVector,
00157     vtkInformationVector *outputVector);
00158 #else
00159 void WriteSlice(vtkImageData *data);
00160 #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00161 int WriteGDCMData(vtkImageData *data, int timeStep);
00162
00163 protected:
00164 virtual /*const*/ char *GetFileName();
00165
00166 private:
00167 vtkGDCMImageWriter(const vtkGDCMImageWriter&); // Not implemented.
00168 void operator=(const vtkGDCMImageWriter&); // Not implemented.
00169
00170 // VTK structs:
00171 //vtkLookupTable *LookupTable;
00172 vtkMedicalImageProperties *MedicalImageProperties;
00173 char *StudyUID;
00174 char *SeriesUID;
00175
00176 int DataUpdateExtent[6];
00177 int ImageFormat;
00178
00179 vtkStringArray *FileNames;
00180 vtkMatrix4x4 *DirectionCosines;
00181
00182 double Shift;
00183 double Scale;
00184 int FileLowerLeft;
00185 int PlanarConfiguration;
00186 int LossyFlag;
00187 int CompressionType;
00188 };
00189
00190 #endif

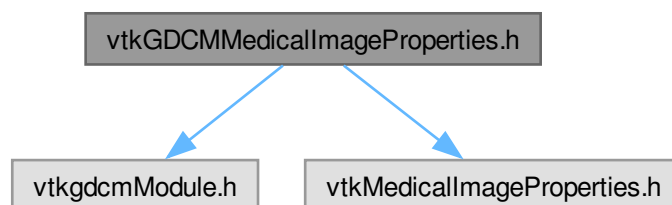
```

13.611 vtkGDCMMedicalImageProperties.h File Reference

#include "vtkgdcmModule.h"

#include "vtkMedicalImageProperties.h"

Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class [vtkGDCMMedicalImageProperties](#)

Namespaces

- namespace [gdcm](#)

13.612 vtkGDCMMedicalImageProperties.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  // .NAME vtkGDCMMedicalImageProperties - some medical image properties.
00015  // .SECTION Description
00016  // vtkGDCMMedicalImageProperties is a helper class that can be used by medical
00017  // image readers and applications to encapsulate medical image/acquisition
00018  // properties. Later on, this should probably be extended to add
00019  // any user-defined property.
00020  // .SECTION See Also
00021  // vtkMedicalImageReader2
00022  00023  #ifndef VTKGDCMMEDICALIMAGEPROPERTIES_H
00024  #define VTKGDCMMEDICALIMAGEPROPERTIES_H
00025  00026  #include "vtkgdcmModule.h"
00027  #include "vtkMedicalImageProperties.h"
00028  00029  class vtkGDCMMedicalImagePropertiesInternals;
00030  //BTX
00031  namespace gdcm { class File; }
00032  //ETX
00033  00034  class VTKGDCM_EXPORT vtkGDCMMedicalImageProperties : public vtkMedicalImageProperties
00035  {
00036  public:
00037    static vtkGDCMMedicalImageProperties *New();
00038    vtkTypeMacro(vtkGDCMMedicalImageProperties,vtkMedicalImageProperties);
00039    void PrintSelf(ostream& os, vtkIndent indent);
00040  00041    // Description:
00042    // Convenience method to reset all fields to an empty string/value
00043    virtual void Clear();
00044  00045  /*
00046    // Description:
00047    // Patient name
00048    // For ex: DICOM (0010,0010) = DOE,JOHN
00049    vtkSetStringMacro(PatientName);
00050    vtkGetStringMacro(PatientName);
00051  00052    // Description:
00053    // Patient ID
00054    // For ex: DICOM (0010,0020) = 1933197
00055    vtkSetStringMacro(PatientID);
00056    vtkGetStringMacro(PatientID);
00057  */

```

```

00058 // Description:
00059 // Patient age
00060 // Format: nnnD, nnW, nnnM or nnnY (eventually nnD, nnW, nnY)
00061 // with D (day), M (month), W (week), Y (year)
00062 // For ex: DICOM (0010,1010) = 031Y
00063 vtkSetStringMacro(PatientAge);
00064 vtkGetStringMacro(PatientAge);
00065
00066 // Description:
00067 // Take as input a string in VR=AS (DICOM PS3.5) and extract either
00068 // different fields namely: year month week day
00069 // Return 0 on error, 1 on success
00070 // One can test fields if they are different from -1 upon success
00071 static int GetAgeAsFields(const char *age, int &year, int &month, int &week, int &day);
00072
00073 // For Tcl:
00074 // From C++ use GetPatientAge + GetAgeAsField
00075 // Those function parse a DICOM string, and return the value of the number expressed
00076 // this is either expressed in year, month or days. Thus if a string is expressed in years
00077 // GetPatientAgeDay/GetPatientAgeWeek/GetPatientAgeMonth will return 0
00078 int GetPatientAgeYear();
00079 int GetPatientAgeMonth();
00080 int GetPatientAgeWeek();
00081 int GetPatientAgeDay();
00082
00083 // Description:
00084 // Patient sex
00085 // For ex: DICOM (0010,0040) = M
00086 vtkSetStringMacro(PatientSex);
00087 vtkGetStringMacro(PatientSex);
00088
00089 // Description:
00090 // Patient birth date
00091 // Format: yyyyymmdd
00092 // For ex: DICOM (0010,0030) = 19680427
00093 vtkSetStringMacro(PatientBirthDate);
00094 vtkGetStringMacro(PatientBirthDate);
00095
00096 // For Tcl:
00097 // From C++ use GetPatientBirthDate + GetDateAsFields
00098 int GetPatientBirthDateYear();
00099 int GetPatientBirthDateMonth();
00100 int GetPatientBirthDateDay();
00101
00102 // Description:
00103 // Study Date
00104 // Format: yyyyymmdd
00105 // For ex: DICOM (0008,0020) = 20030617
00106 vtkSetStringMacro(StudyDate);
00107 vtkGetStringMacro(StudyDate);
00108
00109 // Description:
00110 // Acquisition Date
00111 // Format: yyyyymmdd
00112 // For ex: DICOM (0008,0022) = 20030617
00113 vtkSetStringMacro(AcquisitionDate);
00114 vtkGetStringMacro(AcquisitionDate);
00115
00116 // For Tcl:
00117 // From C++ use GetAcquisitionDate + GetDateAsFields
00118 int GetAcquisitionDateYear();
00119 int GetAcquisitionDateMonth();
00120 int GetAcquisitionDateDay();
00121
00122 // Description:
00123 // Study Time
00124 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00125 // For ex: DICOM (0008,0030) = 162552.0705 or 230012, or 0012
00126 vtkSetStringMacro(StudyTime);
00127 vtkGetStringMacro(StudyTime);
00128
00129 // Description:
00130 // Acquisition time
00131 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00132 // For ex: DICOM (0008,0032) = 162552.0705 or 230012, or 0012
00133 vtkSetStringMacro(AcquisitionTime);
00134 vtkGetStringMacro(AcquisitionTime);
00135
00136 // Description:
00137 // Image Date aka Content Date
00138 // Format: yyyyymmdd

```

```
00139 // For ex: DICOM (0008,0023) = 20030617
00140 vtkSetStringMacro(ImageDate);
00141 vtkGetStringMacro(ImageDate);
00142
00143 // For Tcl:
00144 // From C++ use GetImageDate + GetDateAsFields
00145 int GetImageDateYear();
00146 int GetImageDateMonth();
00147 int GetImageDateDay();
00148
00149 // Description:
00150 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and extract the
00151 // different fields namely: year month day
00152 // Return 0 on error, 1 on success
00153 static int GetDateAsFields(const char *date, int &year, int &month, int &day);
00154
00155 // Description:
00156 // Take as input a string in ISO 8601 date (YYYY/MM/DD) and construct a
00157 // locale date based on the different fields (see GetDateAsFields to extract
00158 // different fields)
00159 // Return 0 on error, 1 on success
00160 static int GetDateAsLocale(const char *date, char *locale);
00161
00162 // Description:
00163 // Image Time
00164 // Format: hhmmss.frac (any trailing component(s) can be omitted)
00165 // For ex: DICOM (0008,0033) = 162552.0705 or 230012, or 0012
00166 vtkSetStringMacro(ImageTime);
00167 vtkGetStringMacro(ImageTime);
00168
00169 // Description:
00170 // Image number
00171 // For ex: DICOM (0020,0013) = 1
00172 vtkSetStringMacro(ImageNumber);
00173 vtkGetStringMacro(ImageNumber);
00174
00175 // Description:
00176 // Series number
00177 // For ex: DICOM (0020,0011) = 902
00178 vtkSetStringMacro(SeriesNumber);
00179 vtkGetStringMacro(SeriesNumber);
00180
00181 // Description:
00182 // Series Description
00183 // User provided description of the Series
00184 // For ex: DICOM (0008,103e) = SCOUT
00185 vtkSetStringMacro(SeriesDescription);
00186 vtkGetStringMacro(SeriesDescription);
00187
00188 // Description:
00189 // Study ID
00190 // For ex: DICOM (0020,0010) = 37481
00191 vtkSetStringMacro(StudyID);
00192 vtkGetStringMacro(StudyID);
00193
00194 // Description:
00195 // Study description
00196 // For ex: DICOM (0008,1030) = BRAIN/C-SP/FACIAL
00197 vtkSetStringMacro(StudyDescription);
00198 vtkGetStringMacro(StudyDescription);
00199
00200 // Description:
00201 // Modality
00202 // For ex: DICOM (0008,0060)= CT
00203 vtkSetStringMacro(Modality);
00204 vtkGetStringMacro(Modality);
00205
00206 // Description:
00207 // Manufacturer
00208 // For ex: DICOM (0008,0070) = Siemens
00209 vtkSetStringMacro(Manufacturer);
00210 vtkGetStringMacro(Manufacturer);
00211
00212 // Description:
00213 // Manufacturer's Model Name
00214 // For ex: DICOM (0008,1090) = LightSpeed QX/i
00215 vtkSetStringMacro(ManufacturerModelName);
00216 vtkGetStringMacro(ManufacturerModelName);
00217
00218 // Description:
00219 // Station Name
```

```
00220 // For ex: DICOM (0008,1010) = LSPD_OC8
00221 vtkSetStringMacro(StationName);
00222 vtkGetStringMacro(StationName);
00223
00224 // Description:
00225 // Institution Name
00226 // For ex: DICOM (0008,0080) = FooCity Medical Center
00227 vtkSetStringMacro(InstitutionName);
00228 vtkGetStringMacro(InstitutionName);
00229
00230 // Description:
00231 // Convolution Kernel (or algorithm used to reconstruct the data)
00232 // For ex: DICOM (0018,1210) = Bone
00233 vtkSetStringMacro(ConvolutionKernel);
00234 vtkGetStringMacro(ConvolutionKernel);
00235
00236 // Description:
00237 // Slice Thickness (Nominal reconstructed slice thickness, in mm)
00238 // For ex: DICOM (0018,0050) = 0.273438
00239 vtkSetStringMacro(SliceThickness);
00240 vtkGetStringMacro(SliceThickness);
00241 virtual double GetSliceThicknessAsDouble();
00242
00243 // Description:
00244 // Peak kilo voltage output of the (x-ray) generator used
00245 // For ex: DICOM (0018,0060) = 120
00246 vtkSetStringMacro(KVP);
00247 vtkGetStringMacro(KVP);
00248
00249 // Description:
00250 // Gantry/Detector tilt (Nominal angle of tilt in degrees of the scanning
00251 // gantry.)
00252 // For ex: DICOM (0018,1120) = 15
00253 vtkSetStringMacro(GantryTilt);
00254 vtkGetStringMacro(GantryTilt);
00255 virtual double GetGantryTiltAsDouble();
00256
00257 // Description:
00258 // Echo Time
00259 // (Time in ms between the middle of the excitation pulse and the peak of
00260 // the echo produced)
00261 // For ex: DICOM (0018,0081) = 105
00262 vtkSetStringMacro(EchoTime);
00263 vtkGetStringMacro(EchoTime);
00264
00265 // Description:
00266 // Echo Train Length
00267 // (Number of lines in k-space acquired per excitation per image)
00268 // For ex: DICOM (0018,0091) = 35
00269 vtkSetStringMacro(EchoTrainLength);
00270 vtkGetStringMacro(EchoTrainLength);
00271
00272 // Description:
00273 // Repetition Time
00274 // The period of time in msec between the beginning of a pulse sequence and
00275 // the beginning of the succeeding (essentially identical) pulse sequence.
00276 // For ex: DICOM (0018,0080) = 2040
00277 vtkSetStringMacro(RepetitionTime);
00278 vtkGetStringMacro(RepetitionTime);
00279
00280 // Description:
00281 // Exposure time (time of x-ray exposure in msec)
00282 // For ex: DICOM (0018,1150) = 5
00283 vtkSetStringMacro(ExposureTime);
00284 vtkGetStringMacro(ExposureTime);
00285
00286 // Description:
00287 // X-ray tube current (in mA)
00288 // For ex: DICOM (0018,1151) = 400
00289 vtkSetStringMacro(XRayTubeCurrent);
00290 vtkGetStringMacro(XRayTubeCurrent);
00291
00292 // Description:
00293 // Exposure (The exposure expressed in mAs, for example calculated
00294 // from Exposure Time and X-ray Tube Current)
00295 // For ex: DICOM (0018,1152) = 114
00296 vtkSetStringMacro(Exposure);
00297 vtkGetStringMacro(Exposure);
00298
00299 // Interface to allow insertion of user define values, for instance in DICOM one would want to
00300 // store the Protocol Name (0018,1030), in this case one would do:
```

```

00301 // AddUserDefinedValue( "Protocol Name", "T1W/SE/1024" );
00302 void AddUserDefinedValue(const char *name, const char *value);
00303 // Get a particular user value
00304 const char *GetUserDefinedValue(const char *name);
00305 // Get the number of user defined values
00306 unsigned int GetNumberOfUserDefinedValues();
00307 // Get a name/value by index
00308 const char *GetUserDefinedNameByIndex(unsigned int idx);
00309 const char *GetUserDefinedValueByIndex(unsigned int idx);
00310
00311 // Description:
00312 // Copy the contents of p to this instance.
00313 virtual void DeepCopy(vtkGDCMMedicalImageProperties *p);
00314
00315 // Description:
00316 // Add/Remove/Query the window/level presets that may have been associated
00317 // to a medical image. Window is also known as 'width', level is also known
00318 // as 'center'. The same window/level pair can not be added twice.
00319 // As a convenience, a comment (aka Explanation) can be associated to a preset.
00320 // For ex: DICOM Window Center (0028,1050) = 00045\000470
00321 //          DICOM Window Width (0028,1051) = 0106\03412
00322 //          DICOM Window Center Width Explanation (0028,1055) = WINDOW1\WINDOW2
00323 virtual void AddWindowLevelPreset(double w, double l);
00324 virtual void RemoveWindowLevelPreset(double w, double l);
00325 virtual void RemoveAllWindowsLevelPresets();
00326 virtual int GetNumberOfWindowLevelPresets();
00327 virtual int HasWindowLevelPreset(double w, double l);
00328 virtual int GetNthWindowLevelPreset(int idx, double *w, double *l);
00329 virtual double* GetNthWindowLevelPreset(int idx);
00330 virtual void SetNthWindowLevelPresetComment(int idx, const char *comment);
00331 virtual const char* GetNthWindowLevelPresetComment(int idx);
00332
00333 // Description:
00334 // Mapping from a sliceidx within a volumeidx into a DICOM Instance UID
00335 // Some DICOM reader can populate this structure so that later on from a slice index
00336 // in a vtkImageData volume we can backtrack and find out which 2d slice it was coming from
00337 const char *GetInstanceUIDFromSliceID(int volumeidx, int sliceid);
00338 void SetInstanceUIDFromSliceID(int volumeidx, int sliceid, const char *uid);
00339
00340 // Description:
00341 // Provides the inverse mapping. Returns -1 if a slice for this uid is
00342 // not found.
00343 int GetSliceIDFromInstanceUID(int &volumeidx, const char *uid);
00344
00345 //BTX
00346 typedef enum {
00347     AXIAL = 0,
00348     CORONAL,
00349     SAGITTAL
00350 } OrientationType;
00351 //ETX
00352 int GetOrientationType(int volumeidx);
00353 void SetOrientationType(int volumeidx, int orientation);
00354 static const char *GetStringFromOrientationType(unsigned int type);
00355 */
00356 protected:
00357     vtkGDCMMedicalImageProperties();
00358     ~vtkGDCMMedicalImageProperties();
00359
00360 //BTX
00361 friend class vtkGDCMImageReader;
00362 friend class vtkGDCMImageReader2;
00363 friend class vtkGDCMImageWriter;
00364 void PushBackFile(gdcm::File const &f);
00365 gdcm::File const & GetFile(unsigned int t);
00366 //ETX
00367
00368 private:
00369     vtkGDCMMedicalImagePropertiesInternals *Internals;
00370
00371     vtkGDCMMedicalImageProperties(const vtkGDCMMedicalImageProperties&); // Not implemented.
00372     void operator=(const vtkGDCMMedicalImageProperties&); // Not implemented.
00373 };
00374
00375 #endif

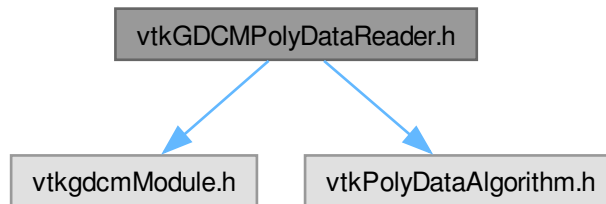
```

13.613 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class [vtkGDCMPolyDataReader](#)

Namespaces

- namespace [gdcm](#)

13.614 vtkGDCMPolyDataReader.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkGDCMPolyDataReader - read DICOM PolyData files (Contour Data...)
00015  // .SECTION Description
00016  // For now only support RTSTRUCT (RT Structure Set Storage)
00017  // .SECTION TODO
00018  // Need to do the same job for DVH Sequence/DVH Data...
00019  // .SECTION Warning
00020  // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
00021  // it is *required* that FileLowerLeft is set to ON as coordinate system
00022  // would be inconsistent in between the two data structures.
00023  //

```

```

00024 // .SECTION See Also
00025 // vtkGDCMImageReader vtkGDCMPolyDataWriter vtkRTStructSetProperties
00026
00027
00028 #ifndef VTKGDCMPOLYDATAREADER_H
00029 #define VTKGDCMPOLYDATAREADER_H
00030
00031 #include "vtkgdcmModule.h"
00032 #include "vtkPolyDataAlgorithm.h"
00033
00034 class vtkMedicalImageProperties;
00035 class vtkRTStructSetProperties;
00036 //BTX
00037 namespace gdcmm { class Reader; }
00038 //ETX
00039 class VTKGDCM_EXPORT vtkGDCMPolyDataReader : public vtkPolyDataAlgorithm
00040 {
00041 public:
00042     static vtkGDCMPolyDataReader *New();
00043     vtkTypeMacro(vtkGDCMPolyDataReader,vtkPolyDataAlgorithm);
00044     virtual void PrintSelf(ostream& os, vtkIndent indent);
00045
00046     // Description:
00047     // Set/Get the filename of the file to be read
00048     vtkSetStringMacro(FileName);
00049     vtkGetStringMacro(FileName);
00050
00051     // Description:
00052     // Get the medical image properties object
00053     vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00054     vtkGetObjectMacro(RTStructSetProperties, vtkRTStructSetProperties);
00055
00056 protected:
00057     vtkGDCMPolyDataReader();
00058     ~vtkGDCMPolyDataReader();
00059
00060     char *FileName;
00061     vtkMedicalImageProperties *MedicalImageProperties;
00062     vtkRTStructSetProperties *RTStructSetProperties;
00063 //BTX
00064 void FillMedicalImageInformation(const gdcmm::Reader &reader);
00065 //ETX
00066
00067 int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00068 int RequestInformation(
00069     vtkInformation *vtkNotUsed(request),
00070     vtkInformationVector **vtkNotUsed(inputVector),
00071     vtkInformationVector *outputVector);
00072 //BTX
00073 int RequestInformation_RTStructureSetStorage(gdcmm::Reader const & reader);
00074 int RequestData_RTStructureSetStorage(gdcmm::Reader const &reader, vtkInformationVector *outputVector);
00075 int RequestInformation_HemodynamicWaveformStorage(gdcmm::Reader const & reader);
00076 int RequestData_HemodynamicWaveformStorage(gdcmm::Reader const &reader, vtkInformationVector *outputVector);
00077 //ETX
00078
00079 private:
00080     vtkGDCMPolyDataReader(const vtkGDCMPolyDataReader&); // Not implemented.
00081     void operator=(const vtkGDCMPolyDataReader&); // Not implemented.
00082 };
00083
00084
00085 #endif

```

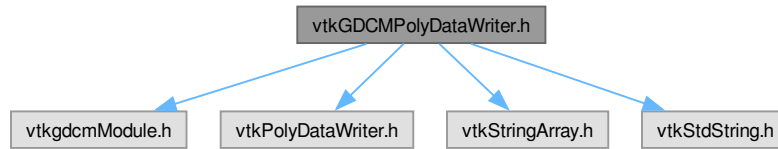
13.615 vtkGDCMPolyDataWriter.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"

```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class [vtkGDCMPolyDataWriter](#)

Namespaces

- namespace [gdcM](#)

13.616 vtkGDCMPolyDataWriter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  // .NAME vtkGDCMPolyDataWriter - writer DICOM PolyData files (Contour Data...)
00015  // .SECTION Description
00016  // For now only support RTSTRUCT (RT Structure Set Storage)
00017  // .SECTION TODO
00018  // Need to do the same job for DVH Sequence/DVH Data...
00019  // .SECTION Warning
00020  //
00021  // .SECTION See Also
00022  // vtkGDCMImageReader vtkGDCMPolyDataReader vtkRTStructSetProperties
00023  00024
00025  #ifndef VTKGDCMPOLYDATAWRITER_H
00026  #define VTKGDCMPOLYDATAWRITER_H
00027
00028  #include "vtkgdcModule.h"
00029  #include "vtkPolyDataWriter.h"
00030  #include "vtkStringArray.h"
00031  #include "vtkStdString.h"
00032
00033
00034  class vtkMedicalImageProperties;
00035  class vtkRTStructSetProperties;
00036  //BTX
  
```



```

00037 namespace gdcmm { class File; }
00038 //ETX
00039 class VTKGDCM_EXPORT vtkGDCMPolyDataWriter : public vtkPolyDataWriter
00040 {
00041 public:
00042 static vtkGDCMPolyDataWriter *New();
00043 vtkTypeMacro(vtkGDCMPolyDataWriter,vtkPolyDataWriter);
00044 virtual void PrintSelf(ostream& os, vtkIndent indent);
00045
00046 // Description:
00047 // Set/Get the filename of the file to be read
00048 // vtkSetStringMacro(FileName);
00049 // vtkGetStringMacro(FileName);
00050
00051 // Description:
00052 // Get the medical image properties object
00053 // vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
00054 virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);
00055
00056 virtual void SetRTStructSetProperties(vtkRTStructSetProperties *pd);
00057
00058
00059 //this function will initialize the contained rtstructset with
00060 //the inputs of the writer and the various extra information
00061 //necessary for writing a complete rtstructset.
00062 //NOTE: inputs must be set BEFORE calling this function!
00063 //NOTE: the number of outputs for the appendpolydata MUST MATCH the ROI vectors!
00064 void InitializeRTStructSet(vtkStdString inDirectory,
00065     vtkStdString inStructLabel, vtkStdString inStructName,
00066     vtkStringArray* inROINames,
00067     vtkStringArray* inROIAlgorithmName,
00068     vtkStringArray* inROIType);
00069
00070 // make parent class public...
00071 void SetNumberOfInputPorts(int n);
00072
00073 protected:
00074 vtkGDCMPolyDataWriter();
00075 ~vtkGDCMPolyDataWriter();
00076
00077 vtkMedicalImageProperties *MedicalImageProperties;
00078 vtkRTStructSetProperties *RTStructSetProperties;
00079
00080 void WriteData();
00081 //BTX
00082 void WriteRTSTRUCTInfo(gdcm::File &file);
00083 void WriteRTSTRUCTData(gdcm::File &file, int num);
00084 //ETX
00085
00086 private:
00087 vtkGDCMPolyDataWriter(const vtkGDCMPolyDataWriter&); // Not implemented.
00088 void operator=(const vtkGDCMPolyDataWriter&); // Not implemented.
00089 };
00090
00091 #endif

```

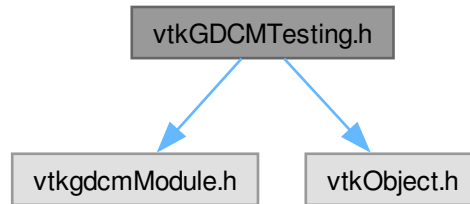
13.617 vtkGDCMTesting.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkObject.h"

```

Include dependency graph for vtkGDCMTesting.h:



Classes

- class [vtkGDCMTesting](#)

13.618 vtkGDCMTesting.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkGDCMTesting - GDCM Testing
00015  // .SECTION Description
00016  // GDCM Testing
00017
00018  // .SECTION See Also
00019  // vtkTesting
00020
00021  #ifndef VTKGDCMTESTING_H
00022  #define VTKGDCMTESTING_H
00023
00024  #include "vtkgdcModule.h"
00025  #include "vtkObject.h"
00026
00027  class VTKGDCM_EXPORT vtkGDCMTesting : public vtkObject
00028  {
00029  public:
00030    static vtkGDCMTesting *New();
00031    vtkTypeMacro(vtkGDCMTesting,vtkObject);
00032    void PrintSelf(ostream& os, vtkIndent indent);
00033
00034    static const char *GetVTKDataRoot();
00035    static const char *GetGDCMDataRoot();
00036
00037  //BTX
00038  typedef const char* const (*MD5MetaImagesType)[3];
  
```

```

00039 static const char * const * GetMD5MetaImage(unsigned int file);
00040 //ETX
00041 static unsigned int GetNumberOfMD5MetaImages();
00042
00043 static const char * GetMHDMD5FromFile(const char *filepath);
00044 static const char * GetRAWMD5FromFile(const char *filepath);
00045
00046 protected:
00047   vtkGDCMTesting();
00048   ~vtkGDCMTesting();
00049
00050 private:
00051   vtkGDCMTesting(const vtkGDCMTesting&); // Not implemented.
00052   void operator=(const vtkGDCMTesting&); // Not implemented.
00053 };
00054
00055 #endif

```

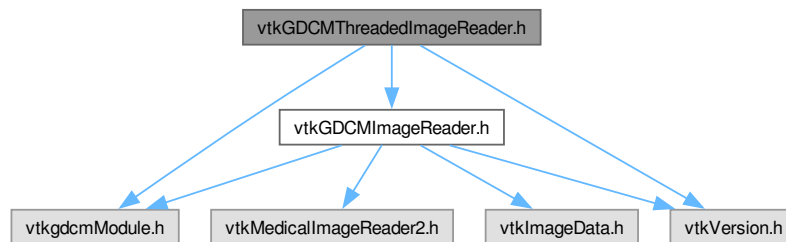
13.619 vtkGDCMThreadedImageReader.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkGDCMImageReader.h"
#include "vtkVersion.h"

```

Include dependency graph for vtkGDCMThreadedImageReader.h:



Classes

- class [vtkGDCMThreadedImageReader](#)

13.620 vtkGDCMThreadedImageReader.h

[Go to the documentation of this file.](#)

```

00001
00002 /*=====
00003 Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005 Copyright (c) 2006-2011 Mathieu Malaterre
00006 All rights reserved.
00007 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009 This software is distributed WITHOUT ANY WARRANTY; without even
00010 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

00011     PURPOSE. See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkGDCMThreadedImageReader - read DICOM files with multiple threads
00015 // .SECTION Description
00016 // vtkGDCMThreadedImageReader is a source object that reads some DICOM files
00017 // This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
00018 // read approx N times faster than when reading in a single thread.
00019 //
00020 // .SECTION Warning: Advanced users only. Do not use this class in the general case,
00021 // you have to understand how physically medium works first (sequential reading for
00022 // instance) before playing with this class
00023 //
00024 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00025 // upside down as VTK would expect, use this option only if you know what you are doing
00026 //
00027 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
00028 //
00029 // .SECTION FIXME: you need to call SetFileName when reading a volume file (multiple slices DICOM)
00030 // since SetFileNames expect each single file to be single slice (see parent class)
00031 //
00032 // .SECTION BUG: you should really consider using vtkGDCMThreadedImageReader2 instead !
00033 //
00034 // .SECTION See Also
00035 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMThreadedImageReader2
00036
00037 #ifndef VTKGDCMTHREADEDIMAGEREADER_H
00038 #define VTKGDCMTHREADEDIMAGEREADER_H
00039
00040 #include "vtkgdcmModule.h"
00041 #include "vtkGDCMImageReader.h"
00042 #include "vtkVersion.h"
00043
00044 class VTKGDCM_EXPORT vtkGDCMThreadedImageReader : public vtkGDCMImageReader
00045 {
00046 public:
00047     static vtkGDCMThreadedImageReader *New();
00048     vtkTypeMacro(vtkGDCMThreadedImageReader,vtkGDCMImageReader);
00049     virtual void PrintSelf(ostream& os, vtkIndent indent);
00050
00051     // Description:
00052     // Explicitly set the Rescale Intercept (0028,1052)
00053     vtkSetMacro(Shift,double);
00054
00055     // Description:
00056     // Explicitly get/set the Rescale Slope (0028,1053)
00057     vtkSetMacro(Scale,double);
00058
00059     // Description:
00060     // Determine whether or not reader should use value from Shift/Scale
00061     // Default is 1
00062     vtkSetMacro(UseShiftScale,int);
00063     vtkGetMacro(UseShiftScale,int);
00064     vtkBooleanMacro(UseShiftScale,int);
00065
00066     // Within this class this is allowed to set the Number of Overlays from outside
00067     //vtkSetMacro(NumberOfOverlays,int);
00068
00069 protected:
00070     vtkGDCMThreadedImageReader();
00071     ~vtkGDCMThreadedImageReader();
00072
00073     #if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
00074     int RequestInformation(vtkInformation *request,
00075                           vtkInformationVector **inputVector,
00076                           vtkInformationVector *outputVector);
00077     int RequestData(vtkInformation *request,
00078                    vtkInformationVector **inputVector,
00079                    vtkInformationVector *outputVector);
00080     #else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00081     void ExecuteInformation();
00082     void ExecuteData(vtkDataObject *out);
00083     #endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
00084
00085     void ReadFiles(unsigned int nfiles, const char *filenames[]);
00086     void RequestDataCompat();
00087
00088 private:
00089     vtkGDCMThreadedImageReader(const vtkGDCMThreadedImageReader&); // Not implemented.
00090     void operator=(const vtkGDCMThreadedImageReader&); // Not implemented.

```

```

00091
00092  int UseShiftScale;
00093 };
00094
00095 #endif

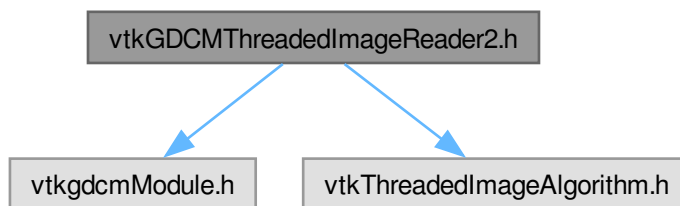
```

13.621 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkgdcModule.h"
```

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



Classes

- class [vtkGDCMThreadedImageReader2](#)

13.622 vtkGDCMThreadedImageReader2.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011   PURPOSE. See the above copyright notice for more information.
00012
00013   =====*/
00014 // .NAME vtkGDCMThreadedImageReader2 - read DICOM files with multiple threads
00015 // .SECTION Description
00016 // vtkGDCMThreadedImageReader2 is a source object that reads some DICOM files
00017 // This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
00018 // read approx N times faster than when reading in a single thread assuming the IO is
00019 // not a bottleneck operation.
00020 // If looking for a single threaded class see: vtkGDCMImageReader
00021 //

```

```

00022 // .SECTION Warning: Advanced users only. Do not use this class in the general case,
00023 // you have to understand how physically medium works first (sequential reading for
00024 // instance) before playing with this class
00025 //
00026 // .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
00027 // upside down as VTK would expect, use this option only if you know what you are doing
00028 //
00029 // .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
00030 //
00031 // .SECTION FIXME: need to implement reading of series of 3D files
00032 //
00033 // .SECTION Implementation note: this class is meant to supersede vtkGDCMThreadedImageReader
00034 // because it had support for ProgressEvent support even from python layer. There is a
00035 // subtle trick down in the threading mechanism in VTK were the main thread (talking to the
00036 // python interpreter) is also part of the execution process (and the N-1 other thread
00037 // are just there to execute the remaining of ThreadedRequestData), this separation into
00038 // two types of thread is necessary to achieve a working implementation of UpdateProgress
00039 //
00040 // .SECTION See Also
00041 // vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMImageReader
00042 //
00043 #ifndef VTKGDCMTHREADEDIMAGEREADER2_H
00044 #define VTKGDCMTHREADEDIMAGEREADER2_H
00045 //
00046 #include "vtkgdcModule.h"
00047 #include "vtkThreadedImageAlgorithm.h"
00048 //
00049 class vtkStringArray;
00050 class VTKGDCM_EXPORT vtkGDCMThreadedImageReader2 : public vtkThreadedImageAlgorithm
00051 {
00052 public:
00053     static vtkGDCMThreadedImageReader2 *New();
00054     vtkTypeMacro(vtkGDCMThreadedImageReader2,vtkThreadedImageAlgorithm);
00055     virtual void PrintSelf(ostream& os, vtkIndent indent);
00056 //
00057     vtkGetMacro(FileLowerLeft,int);
00058     vtkSetMacro(FileLowerLeft,int);
00059     vtkBooleanMacro(FileLowerLeft,int);
00060 //
00061     vtkGetMacro(NumberOfOverlays,int);
00062 //
00063     vtkSetMacro(DataScalarType,int);
00064     vtkGetMacro(DataScalarType,int);
00065 //
00066     vtkSetMacro(NumberOfScalarComponents,int);
00067     vtkGetMacro(NumberOfScalarComponents,int);
00068 //
00069     vtkGetMacro(LoadOverlays,int);
00070     vtkSetMacro(LoadOverlays,int);
00071     vtkBooleanMacro(LoadOverlays,int);
00072 //
00073     vtkSetVector6Macro(DataExtent,int);
00074     vtkGetVector6Macro(DataExtent,int);
00075 //
00076     vtkSetVector3Macro(DataOrigin,double);
00077     vtkGetVector3Macro(DataOrigin,double);
00078 //
00079     vtkSetVector3Macro(DataSpacing,double);
00080     vtkGetVector3Macro(DataSpacing,double);
00081 //
00082     //vtkGetStringMacro(FileName);
00083     //vtkSetStringMacro(FileName);
00084     virtual const char *GetFileName(int i = 0);
00085     virtual void SetFileName(const char *filename);
00086 //
00087     virtual void SetFileNames(vtkStringArray*);
00088     vtkGetObjectMacro(FileNames, vtkStringArray);
00089 //
00090     int SplitExtent(int splitExt[6], int startExt[6],
00091                     int num, int total);
00092 //
00093     // Description:
00094     // Explicitly set the Rescale Intercept (0028,1052)
00095     vtkSetMacro(Shift,double);
00096     vtkGetMacro(Shift,double);
00097 //
00098     // Description:
00099     // Explicitly get/set the Rescale Slope (0028,1053)
00100     vtkSetMacro(Scale,double);
00101     vtkGetMacro(Scale,double);
00102 //

```

```

00103 // Description:
00104 // Determine whether or not reader should use value from Shift/Scale
00105 // Default is 1
00106 vtkSetMacro(UseShiftScale,int);
00107 vtkGetMacro(UseShiftScale,int);
00108 vtkBooleanMacro(UseShiftScale,int);
00109
00110 protected:
00111   vtkGDCMThreadedImageReader2();
00112   ~vtkGDCMThreadedImageReader2();
00113
00114   int RequestInformation(vtkInformation *request,
00115                         vtkInformationVector **inputVector,
00116                         vtkInformationVector *outputVector);
00117
00118 protected:
00119   void ThreadedRequestData (
00120     vtkInformation * request,
00121     vtkInformationVector** inputVector,
00122     vtkInformationVector * outputVector,
00123     vtkImageData ***inData,
00124     vtkImageData **outData,
00125     int outExt[6], int id);
00126
00127 private:
00128   int FileLowerLeft;
00129   char *FileName;
00130   vtkStringArray *FileNames;
00131   int LoadIconImage;
00132   int DataExtent[6];
00133   int LoadOverlays;
00134   int NumberOfOverlays;
00135   int DataScalarType;
00136
00137   int NumberOfScalarComponents;
00138   double DataSpacing[3];
00139   double DataOrigin[3];
00140   int IconImageDataExtent[6];
00141
00142   double Shift;
00143   double Scale;
00144   int UseShiftScale;
00145
00146 private:
00147   vtkGDCMThreadedImageReader2(const vtkGDCMThreadedImageReader2&); // Not implemented.
00148   void operator=(const vtkGDCMThreadedImageReader2&); // Not implemented.
00149   };
00150
00151 #endif

```

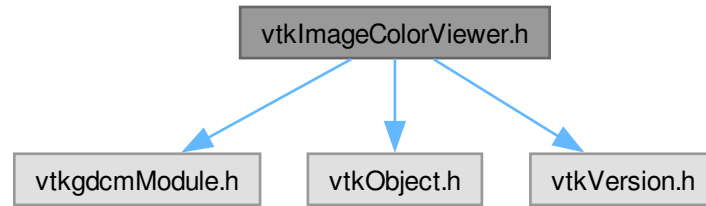
13.623 vtkImageColorViewer.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkObject.h"
#include "vtkVersion.h"

```

Include dependency graph for vtkImageColorViewer.h:



Classes

- class [vtkImageColorViewer](#)

13.624 vtkImageColorViewer.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014  // .NAME vtkImageColorViewer - Display a 2D image.
00015  // .SECTION Description
00016  // vtkImageColorViewer is a convenience class for displaying a 2D image. It
00017  // packages up the functionality found in vtkRenderWindow, vtkRenderer,
00018  // vtkImageActor and vtkImageMapToWindowLevelColors into a single easy to use
00019  // class. This class also creates an image interactor style
00020  // (vtkInteractorStyleImage) that allows zooming and panning of images, and
00021  // supports interactive window/level operations on the image. Note that
00022  // vtkImageColorViewer is simply a wrapper around these classes.
00023  //
00024  // vtkImageColorViewer uses the 3D rendering and texture mapping engine
00025  // to draw an image on a plane. This allows for rapid rendering,
00026  // zooming, and panning. The image is placed in the 3D scene at a
00027  // depth based on the z-coordinate of the particular image slice. Each
00028  // call to SetSlice() changes the image data (slice) displayed AND
00029  // changes the depth of the displayed slice in the 3D scene. This can
00030  // be controlled by the AutoAdjustCameraClippingRange ivar of the
00031  // InteractorStyle member.
00032  //
00033  // It is possible to mix images and geometry, using the methods:
00034  //
00035  // viewer->SetInput( myImage );
00036  // viewer->GetRenderer()->AddActor( myActor );
00037  //
00038  // This can be used to annotate an image with a PolyData of "edges" or

```



```

00039 // or highlight sections of an image or display a 3D isosurface
00040 // with a slice from the volume, etc. Any portions of your geometry
00041 // that are in front of the displayed slice will be visible; any
00042 // portions of your geometry that are behind the displayed slice will
00043 // be obscured. A more general framework (with respect to viewing
00044 // direction) for achieving this effect is provided by the
00045 // vtkImagePlaneWidget .
00046 //
00047 // Note that pressing 'r' will reset the window/level and pressing
00048 // shift+'r' or control+'r' will reset the camera.
00049 //
00050 // .SECTION See Also
00051 // vtkRenderWindow vtkRenderer vtkImageActor vtkImageMapToWindowLevelColors
00052
00053 #ifndef VTKIMAGECOLORVIEWER_H
00054 #define VTKIMAGECOLORVIEWER_H
00055
00056 #include "vtkgdcmModule.h"
00057 #include "vtkObject.h"
00058 #include "vtkVersion.h"
00059
00060 class vtkAlgorithm;
00061 class vtkAlgorithmOutput;
00062 class vtkImageActor;
00063 class vtkImageData;
00064 class vtkImageMapToWindowLevelColors2;
00065 class vtkInformation;
00066 class vtkInteractorStyleImage;
00067 class vtkRenderWindow;
00068 class vtkRenderer;
00069 class vtkRenderWindowInteractor;
00070 class vtkPolyData;
00071
00072 class VTKGDCM_EXPORT vtkImageColorViewer : public vtkObject
00073 {
00074 public:
00075     static vtkImageColorViewer *New();
00076     vtkTypeMacro(vtkImageColorViewer,vtkObject);
00077     void PrintSelf(ostream& os, vtkIndent indent);
00078
00079     // Description:
00080     // Get the name of rendering window.
00081     virtual const char *GetWindowName();
00082
00083     // Description:
00084     // Render the resulting image.
00085     virtual void Render(void);
00086
00087     // Description:
00088     // Set/Get the input image to the viewer.
00089     #if (VTK_MAJOR_VERSION >= 6)
00090     virtual void SetInputData(vtkImageData *in);
00091     #else
00092     virtual void SetInput(vtkImageData *in);
00093     #endif
00094     virtual vtkImageData *GetInput();
00095     virtual void SetInputConnection(vtkAlgorithmOutput* input);
00096     virtual void AddInputConnection(vtkAlgorithmOutput* input);
00097     virtual void AddInput(vtkImageData * input);
00098     //virtual void AddInput(vtkPolyData * input);
00099
00100     double GetOverlayVisibility();
00101     void SetOverlayVisibility(double vis);
00102
00103     // Description:
00104     // Set/get the slice orientation
00105     //BTX
00106     enum
00107     {
00108         SLICE_ORIENTATION_YZ = 0,
00109         SLICE_ORIENTATION_XZ = 1,
00110         SLICE_ORIENTATION_XY = 2
00111     };
00112     //ETX
00113     vtkGetMacro(SliceOrientation, int);
00114     virtual void SetSliceOrientation(int orientation);
00115     virtual void SetSliceOrientationToXY()
00116     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XY); };
00117     virtual void SetSliceOrientationToYZ()
00118     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_YZ); };
00119     virtual void SetSliceOrientationToXZ()

```

```

00120     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XZ); };
00121
00122     // Description:
00123     // Set/Get the current slice to display (depending on the orientation
00124     // this can be in X, Y or Z).
00125     vtkGetMacro(Slice, int);
00126     virtual void SetSlice(int s);
00127
00128     // Description:
00129     // Update the display extent manually so that the proper slice for the
00130     // given orientation is displayed. It will also try to set a
00131     // reasonable camera clipping range.
00132     // This method is called automatically when the Input is changed, but
00133     // most of the time the input of this class is likely to remain the same,
00134     // i.e. connected to the output of a filter, or an image reader. When the
00135     // input of this filter or reader itself is changed, an error message might
00136     // be displayed since the current display extent is probably outside
00137     // the new whole extent. Calling this method will ensure that the display
00138     // extent is reset properly.
00139     virtual void UpdateDisplayExtent();
00140
00141     // Description:
00142     // Return the minimum and maximum slice values (depending on the orientation
00143     // this can be in X, Y or Z).
00144     virtual int GetSliceMin();
00145     virtual int GetSliceMax();
00146     virtual void GetSliceRange(int range[2])
00147     { this->GetSliceRange(range[0], range[1]); }
00148     virtual void GetSliceRange(int &min, int &max);
00149     virtual int* GetSliceRange();
00150
00151     // Description:
00152     // Set window and level for mapping pixels to colors.
00153     virtual double GetColorWindow();
00154     virtual double GetColorLevel();
00155     virtual void SetColorWindow(double s);
00156     virtual void SetColorLevel(double s);
00157
00158     // Description:
00159     // These are here when using a Tk window.
00160     virtual void SetDisplayId(void *a);
00161     virtual void SetWindowId(void *a);
00162     virtual void SetParentId(void *a);
00163
00164     // Description:
00165     // Set/Get the position in screen coordinates of the rendering window.
00166     virtual int* GetPosition();
00167     virtual void SetPosition(int a,int b);
00168     virtual void SetPosition(int a[2]) { this->SetPosition(a[0],a[1]); }
00169
00170     // Description:
00171     // Set/Get the size of the window in screen coordinates in pixels.
00172     virtual int* GetSize();
00173     virtual void SetSize(int a, int b);
00174     virtual void SetSize(int a[2]) { this->SetSize(a[0],a[1]); }
00175
00176     // Description:
00177     // Get the internal render window, renderer, image actor, and
00178     // image map instances.
00179     vtkGetObjectMacro(RenderWindow,vtkRenderWindow);
00180     vtkGetObjectMacro(Renderer, vtkRenderer);
00181     vtkGetObjectMacro(ImageActor,vtkImageActor);
00182     vtkGetObjectMacro(WindowLevel,vtkImageMapToWindowLevelColors2);
00183     vtkGetObjectMacro(InteractorStyle,vtkInteractorStyleImage);
00184
00185     // Description:
00186     // Set your own renderwindow and renderer
00187     virtual void SetRenderWindow(vtkRenderWindow *arg);
00188     virtual void SetRenderer(vtkRenderer *arg);
00189
00190     // Description:
00191     // Attach an interactor for the internal render window.
00192     virtual void SetupInteractor(vtkRenderWindowInteractor*);
00193
00194     // Description:
00195     // Create a window in memory instead of on the screen. This may not
00196     // be supported for every type of window and on some windows you may
00197     // need to invoke this prior to the first render.
00198     virtual void SetOffScreenRendering(int);
00199     virtual int GetOffScreenRendering();
00200     vtkBooleanMacro(OffScreenRendering,int);

```

```

00201
00202 protected:
00203   vtkImageColorViewer();
00204   ~vtkImageColorViewer();
00205
00206   virtual void InstallPipeline();
00207   virtual void UnInstallPipeline();
00208
00209   vtkImageMapToWindowLevelColors2 *WindowLevel;
00210   vtkRenderWindow                *RenderWindow;
00211   vtkRenderer                    *Renderer;
00212   vtkImageActor                  *ImageActor;
00213   vtkImageActor                  *OverlayImageActor;
00214   vtkRenderWindowInteractor      *Interactor;
00215   vtkInteractorStyleImage        *InteractorStyle;
00216
00217   int SliceOrientation;
00218   int FirstRender;
00219   int Slice;
00220
00221   virtual void UpdateOrientation();
00222
00223   #if (VTK_MAJOR_VERSION >= 6)
00224   vtkAlgorithm* GetInputAlgorithm();
00225   vtkInformation* GetInputInformation();
00226   #endif
00227
00228   friend class vtkImageColorViewerCallback;
00229
00230 private:
00231   vtkImageColorViewer(const vtkImageColorViewer&); // Not implemented.
00232   void operator=(const vtkImageColorViewer&); // Not implemented.
00233 };
00234
00235 #endif

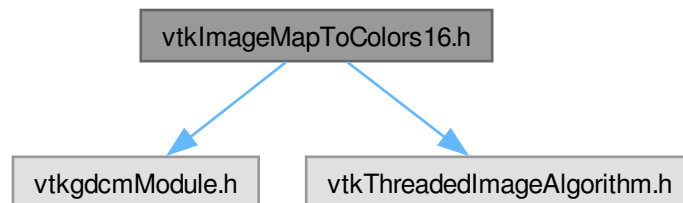
```

13.625 vtkImageMapToColors16.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



Classes

- class [vtkImageMapToColors16](#)

13.626 vtkImageMapToColors16.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  /*=====
00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018  Program: Visualization Toolkit
00019  Module:   $RCSfile: vtkImageMapToColors16.h,v $
00020
00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 // .NAME vtkImageMapToColors16 - map the input image through a lookup table
00031 // .SECTION Description
00032 // The vtkImageMapToColors16 filter will take an input image of any valid
00033 // scalar type, and map the first component of the image through a
00034 // lookup table. The result is an image of type VTK_UNSIGNED_CHAR.
00035 // If the lookup table is not set, or is set to NULL, then the input
00036 // data will be passed through if it is already of type VTK_UNSIGNED_CHAR.
00037
00038 // .SECTION See Also
00039 // vtkLookupTable vtkScalarsToColors
00040
00041 #ifndef VTKIMAGEMAPTOCOLORS16_H
00042 #define VTKIMAGEMAPTOCOLORS16_H
00043
00044
00045 #include "vtkgdcmModule.h"
00046 #include "vtkThreadedImageAlgorithm.h"
00047
00048 class vtkScalarsToColors;
00049
00050 class VTKGDCM_EXPORT vtkImageMapToColors16 : public vtkThreadedImageAlgorithm
00051 {
00052 public:
00053     static vtkImageMapToColors16 *New();
00054     vtkTypeMacro(vtkImageMapToColors16,vtkThreadedImageAlgorithm);
00055     void PrintSelf(ostream& os, vtkIndent indent);
00056
00057     // Description:
00058     // Set the lookup table.
00059     virtual void SetLookupTable(vtkScalarsToColors*);
00060     vtkGetObjectMacro(LookupTable,vtkScalarsToColors);
00061
00062     // Description:
00063     // Set the output format, the default is RGBA.
00064     vtkSetMacro(OutputFormat,int);
00065     vtkGetMacro(OutputFormat,int);
00066     void SetOutputFormatToRGBA() { this->OutputFormat = VTK_RGBA; };
00067     void SetOutputFormatToRGB() { this->OutputFormat = VTK_RGB; };
00068     void SetOutputFormatToLuminanceAlpha() { this->OutputFormat = VTK_LUMINANCE_ALPHA; };
00069     void SetOutputFormatToLuminance() { this->OutputFormat = VTK_LUMINANCE; };
00070
00071     // Description:

```

```

00072 // Set the component to map for multi-component images (default: 0)
00073 vtkSetMacro(ActiveComponent,int);
00074 vtkGetMacro(ActiveComponent,int);
00075
00076 // Description:
00077 // Use the alpha component of the input when computing the alpha component
00078 // of the output (useful when converting monochrome+alpha data to RGBA)
00079 vtkSetMacro(PassAlphaToOutput,int);
00080 vtkBooleanMacro(PassAlphaToOutput,int);
00081 vtkGetMacro(PassAlphaToOutput,int);
00082
00083 // Description:
00084 // We need to check the modified time of the lookup table too.
00085 #ifndef VTK_HAS_MTIME_TYPE
00086 virtual vtkMTimeType GetMTime();
00087 #else
00088 virtual unsigned long GetMTime();
00089 #endif
00090
00091 protected:
00092 vtkImageMapToColors16();
00093 ~vtkImageMapToColors16();
00094
00095 virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00096
00097 void ThreadedRequestData(vtkInformation *request,
00098                          vtkInformationVector **inputVector,
00099                          vtkInformationVector *outputVector,
00100                          vtkImageData ***inData, vtkImageData **outData,
00101                          int extent[6], int id);
00102
00103 virtual int RequestData(vtkInformation *request,
00104                        vtkInformationVector **inputVector,
00105                        vtkInformationVector *outputVector);
00106
00107 vtkScalarsToColors *LookupTable;
00108 int OutputFormat;
00109
00110 int ActiveComponent;
00111 int PassAlphaToOutput;
00112
00113 int DataWasPassed;
00114 private:
00115 vtkImageMapToColors16(const vtkImageMapToColors16&); // Not implemented.
00116 void operator=(const vtkImageMapToColors16&); // Not implemented.
00117 };
00118
00119 #endif

```

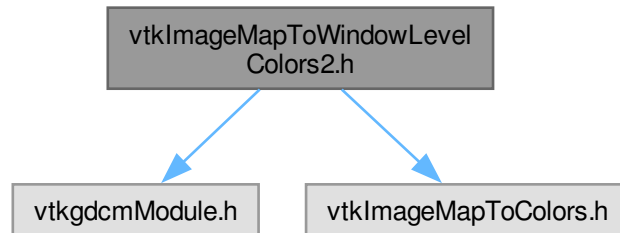
13.627 vtkImageMapToWindowLevelColors2.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkImageMapToColors.h"

```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



Classes

- class [vtkImageMapToWindowLevelColors2](#)

13.628 vtkImageMapToWindowLevelColors2.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  /*=====
00015  00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017  00018  Program: Visualization Toolkit
00019  Module:   $RCSfile: vtkImageMapToWindowLevelColors2.h,v $
00020  00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024  00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028  00029  =====*/
00030  // .NAME vtkImageMapToWindowLevelColors2 - map the input image through a lookup table and window / level it
00031  // .SECTION Description
00032  // The vtkImageMapToWindowLevelColors2 filter will take an input image of any
00033  // valid scalar type, and map the first component of the image through a
00034  // lookup table. This resulting color will be modulated with value obtained

```

```

00035 // by a window / level operation. The result is an image of type
00036 // VTK_UNSIGNED_CHAR. If the lookup table is not set, or is set to NULL, then
00037 // the input data will be passed through if it is already of type
00038 // UNSIGNED_CHAR.
00039 //
00040 // .SECTION See Also
00041 // vtkLookupTable vtkScalarsToColors
00042
00043 #ifndef VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
00044 #define VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
00045
00046 #include "vtkgdcmModule.h"
00047 #include "vtkImageMapToColors.h"
00048
00049 class VTKGDCM_EXPORT vtkImageMapToWindowLevelColors2 : public vtkImageMapToColors
00050 {
00051 public:
00052     static vtkImageMapToWindowLevelColors2 *New();
00053     vtkTypeMacro(vtkImageMapToWindowLevelColors2,vtkImageMapToColors);
00054     void PrintSelf(ostream& os, vtkIndent indent);
00055
00056     // Description:
00057     // Set / Get the Window to use -> modulation will be performed on the
00058     // color based on (S - (L - W/2))/W where S is the scalar value, L is
00059     // the level and W is the window.
00060     vtkSetMacro( Window, double );
00061     vtkGetMacro( Window, double );
00062
00063     // Description:
00064     // Set / Get the Level to use -> modulation will be performed on the
00065     // color based on (S - (L - W/2))/W where S is the scalar value, L is
00066     // the level and W is the window.
00067     vtkSetMacro( Level, double );
00068     vtkGetMacro( Level, double );
00069
00070 protected:
00071     vtkImageMapToWindowLevelColors2();
00072     ~vtkImageMapToWindowLevelColors2();
00073
00074     virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00075     void ThreadedRequestData(vtkInformation *request,
00076                             vtkInformationVector **inputVector,
00077                             vtkInformationVector *outputVector,
00078                             vtkImageData ***inData, vtkImageData **outData,
00079                             int extent[6], int id);
00080     virtual int RequestData(vtkInformation *request,
00081                             vtkInformationVector **inputVector,
00082                             vtkInformationVector *outputVector);
00083
00084     double Window;
00085     double Level;
00086
00087 private:
00088     vtkImageMapToWindowLevelColors2(const vtkImageMapToWindowLevelColors2&); // Not implemented.
00089     void operator=(const vtkImageMapToWindowLevelColors2&); // Not implemented.
00090 };
00091
00092 #endif

```

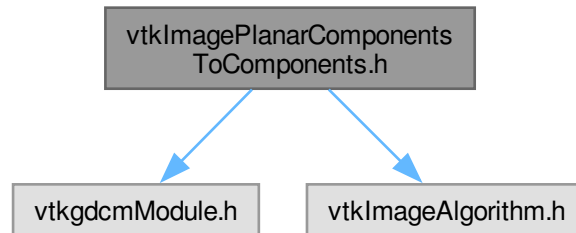
13.629 vtkImagePlanarComponentsToComponents.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkImageAlgorithm.h"

```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



Classes

- class [vtkImagePlanarComponentsToComponents](#)

13.630 vtkImagePlanarComponentsToComponents.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.
00008  00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013  =====*/
00014  /*=====
00015  00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017  00018  Program: Visualization Toolkit
00019  Module:   $RCSfile: vtkImagePlanarComponentsToComponents.h,v $
00020  00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024  00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028  00029  =====*/
00030  // .NAME vtkImagePlanarComponentsToComponents - Converts planar comp to pixel comp
00031  // .SECTION Description
00032  00033  // .SECTION See Also
00034  // TODO: Can I make this filter threaded ?
  
```



```

00035 // TODO: How do I handle the VTK-flipping (FileLowerLeft)?
00036
00037 #ifndef VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
00038 #define VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
00039
00040 #include "vtkgdcmModule.h"
00041 #include "vtkImageAlgorithm.h"
00042
00043 // everything is now handled within the vtkGDCMImageReader as Planar Configuration can not
00044 // be externalized (conflict with file lower left)
00045
00046 #error do not use this class
00047
00048 //class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkThreadedImageAlgorithm
00049 class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkImageAlgorithm
00050 {
00051 public:
00052     static vtkImagePlanarComponentsToComponents *New();
00053     //vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkThreadedImageAlgorithm);
00054     vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkImageAlgorithm);
00055
00056     void PrintSelf(ostream& os, vtkIndent indent);
00057
00058 protected:
00059     vtkImagePlanarComponentsToComponents();
00060     ~vtkImagePlanarComponentsToComponents() {};
00061
00062     // void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00063     //                         int ext[6], int id);
00064     // virtual int RequestInformation (vtkInformation *, vtkInformationVector**, vtkInformationVector *);
00065     virtual int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
00066
00067 private:
00068     vtkImagePlanarComponentsToComponents(const vtkImagePlanarComponentsToComponents&); // Not implemented.
00069     void operator=(const vtkImagePlanarComponentsToComponents&); // Not implemented.
00070 };
00071
00072 #endif

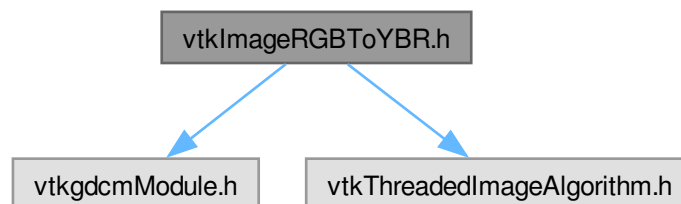
```

13.631 vtkImageRGBToYBR.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



Classes

- class [vtkImageRGBToYBR](#)

13.632 vtkImageRGBToYBR.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  /*=====
00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018  Program: Visualization Toolkit
00019  Module: $RCSfile: vtkImageRGBToYBR.h,v $
00020
00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 // .NAME vtkImageRGBToYBR - Converts YBR components to RGB.
00031 // .SECTION Description
00032 // For each pixel with hue, saturation and value components this filter
00033 // outputs the color coded as red, green, blue. Output type must be the same
00034 // as input type.
00035
00036 // .SECTION See Also
00037 // vtkImageRGBToHSV
00038
00039 #ifndef VTKIMAGERGBTOYBR_H
00040 #define VTKIMAGERGBTOYBR_H
00041
00042 #include "vtkgdcmModule.h"
00043 #include "vtkThreadedImageAlgorithm.h"
00044
00045 class VTKGDCM_EXPORT vtkImageRGBToYBR : public vtkThreadedImageAlgorithm
00046 {
00047 public:
00048     static vtkImageRGBToYBR *New();
00049     vtkTypeMacro(vtkImageRGBToYBR,vtkThreadedImageAlgorithm);
00050
00051     void PrintSelf(ostream& os, vtkIndent indent);
00052
00053 protected:
00054     vtkImageRGBToYBR();
00055     ~vtkImageRGBToYBR() {};
00056
00057     void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00058                          int ext[6], int id);
00059 private:
00060     vtkImageRGBToYBR(const vtkImageRGBToYBR&); // Not implemented.
00061     void operator=(const vtkImageRGBToYBR&); // Not implemented.
00062 };
00063
00064 #endif

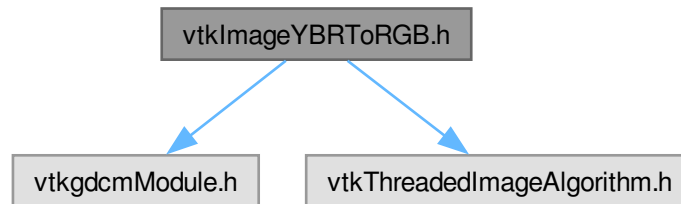
```

13.633 vtkImageYBRToRGB.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



Classes

- class [vtkImageYBRToRGB](#)

13.634 vtkImageYBRToRGB.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004  00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  00006  All rights reserved.
00007  00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008  00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012  00013
00014  =====*/
00015  /*=====
00016  00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017  00017
00018  Program: Visualization Toolkit
00019  Module:   $RCSfile: vtkImageYBRToRGB.h,v $
00020  00020
00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024  00024
00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028  00028

```

```

00029 =====*/
00030 // .NAME vtkImageYBRToRGB - Converts YBR components to RGB.
00031 // .SECTION Description
00032 // For each pixel with hue, saturation and value components this filter
00033 // outputs the color coded as red, green, blue. Output type must be the same
00034 // as input type.
00035
00036 // .SECTION See Also
00037 // vtkImageRGBToHSV
00038
00039 #ifndef VTKIMAGEYBRTORGB_H
00040 #define VTKIMAGEYBRTORGB_H
00041
00042 #include "vtkgdcmModule.h"
00043 #include "vtkThreadedImageAlgorithm.h"
00044
00045 class VTKGDCM_EXPORT vtkImageYBRToRGB : public vtkThreadedImageAlgorithm
00046 {
00047 public:
00048     static vtkImageYBRToRGB *New();
00049     vtkTypeMacro(vtkImageYBRToRGB,vtkThreadedImageAlgorithm);
00050
00051     void PrintSelf(ostream& os, vtkIndent indent);
00052
00053 protected:
00054     vtkImageYBRToRGB();
00055     ~vtkImageYBRToRGB() {};
00056
00057     void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
00058                          int ext[6], int id);
00059 private:
00060     vtkImageYBRToRGB(const vtkImageYBRToRGB&); // Not implemented.
00061     void operator=(const vtkImageYBRToRGB&); // Not implemented.
00062 };
00063
00064 #endif

```

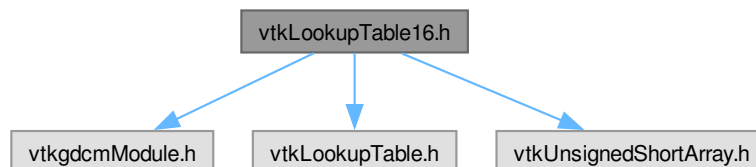
13.635 vtkLookupTable16.h File Reference

```

#include "vtkgdcmModule.h"
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"

```

Include dependency graph for vtkLookupTable16.h:



Classes

- class [vtkLookupTable16](#)

13.636 vtkLookupTable16.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002
00003  Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005  Copyright (c) 2006-2011 Mathieu Malaterre
00006  All rights reserved.
00007  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009  This software is distributed WITHOUT ANY WARRANTY; without even
00010  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011  PURPOSE. See the above copyright notice for more information.
00012
00013  =====*/
00014
00015  /*=====
00016  Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018  Program: Visualization Toolkit
00019  Module: $RCSfile: vtkLookupTable16.h,v $
00020
00021  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022  All rights reserved.
00023  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.
00024
00025  This software is distributed WITHOUT ANY WARRANTY; without even
00026  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00027  PURPOSE. See the above copyright notice for more information.
00028
00029  =====*/
00030 // .NAME vtkLookupTable16 -
00031 // .SECTION Description
00032 //
00033 // .SECTION Caveats
00034 //
00035 // .SECTION See Also
00036 // vtkLookupTable
00037
00038 #ifndef VTKLOOKUPTABLE16_H
00039 #define VTKLOOKUPTABLE16_H
00040
00041 #include "vtkgdcmModule.h"
00042 #include "vtkLookupTable.h"
00043 #include "vtkUnsignedShortArray.h"
00044
00045 class VTKGDCM_EXPORT vtkLookupTable16 : public vtkLookupTable
00046 {
00047 public:
00048     static vtkLookupTable16 *New();
00049
00050     vtkTypeMacro(vtkLookupTable16,vtkLookupTable);
00051     void PrintSelf(ostream& os, vtkIndent indent);
00052
00053     void Build();
00054
00055     void SetNumberOfTableValues(vtkIdType number);
00056
00057     unsigned char *WritePointer(const vtkIdType id, const int number);
00058
00059     unsigned short *GetPointer(const vtkIdType id) {
00060         return this->Table16->GetPointer(4*id); };
00061
00062 protected:
00063     vtkLookupTable16(int size=256, int ext=256);
00064     ~vtkLookupTable16();
00065
00066     vtkUnsignedShortArray *Table16;
00067
00068     void MapScalarsThroughTable2(void *input,
00069                                 unsigned char *output,
00070                                 int inputDataType,
00071                                 int numberOfValues,

```

```

00072             int inputIncrement,
00073             int outputFormat);
00074
00075 private:
00076     vtkLookupTable16(const vtkLookupTable16&); // Not implemented.
00077     void operator=(const vtkLookupTable16&); // Not implemented.
00078 };
00079
00080 //-----
00081 inline unsigned char *vtkLookupTable16::WritePointer(const vtkIdType id,
00082                                                     const int number)
00083 {
00084     //this->InsertTime.Modified();
00085     return (unsigned char*)this->Table16->WritePointer(4*id,4*number);
00086 }
00087
00088 #endif

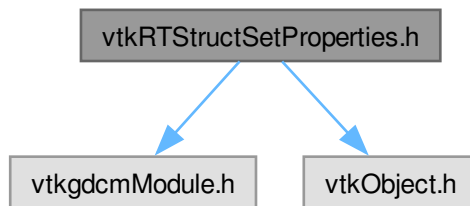
```

13.637 vtkRTStructSetProperties.h File Reference

```
#include "vtkgdcmModule.h"
```

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class [vtkRTStructSetProperties](#)

13.638 vtkRTStructSetProperties.h

[Go to the documentation of this file.](#)

```

00001
00002  /*=====
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009   This software is distributed WITHOUT ANY WARRANTY; without even
00010   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

00011     PURPOSE.  See the above copyright notice for more information.
00012
00013     =====*/
00014 // .NAME vtkRTStructSetProperties - some rtstruct properties.
00015 // .SECTION Description
00016 //
00017 // .SECTION See Also
00018 // vtkGDCMPolyDataReader vtkGDCMPolyDataWriter
00019
00020 #ifndef VTKRTSTRUCTSETPROPERTIES_H
00021 #define VTKRTSTRUCTSETPROPERTIES_H
00022
00023 #include "vtkgdcmModule.h"
00024 #include "vtkObject.h"
00025
00026 class vtkRTStructSetPropertiesInternals;
00027
00028 class VTKGDCM_EXPORT vtkRTStructSetProperties : public vtkObject
00029 {
00030 public:
00031     static vtkRTStructSetProperties *New();
00032     vtkTypeMacro(vtkRTStructSetProperties,vtkObject);
00033     void PrintSelf(ostream& os, vtkIndent indent);
00034
00035     // Description:
00036     // Convenience method to reset all fields to an empty string/value
00037     virtual void Clear();
00038
00039     // Description:
00040     //
00041     vtkSetStringMacro(StructureSetLabel);
00042     vtkGetStringMacro(StructureSetLabel);
00043
00044     vtkSetStringMacro(StructureSetName);
00045     vtkGetStringMacro(StructureSetName);
00046
00047     vtkSetStringMacro(StructureSetDate);
00048     vtkGetStringMacro(StructureSetDate);
00049
00050     vtkSetStringMacro(StructureSetTime);
00051     vtkGetStringMacro(StructureSetTime);
00052
00053     vtkSetStringMacro(SOPInstanceUID);
00054     vtkGetStringMacro(SOPInstanceUID);
00055
00056     vtkSetStringMacro(StudyInstanceUID);
00057     vtkGetStringMacro(StudyInstanceUID);
00058
00059     vtkSetStringMacro(SeriesInstanceUID);
00060     vtkGetStringMacro(SeriesInstanceUID);
00061
00062     vtkSetStringMacro(ReferenceSeriesInstanceUID);
00063     vtkGetStringMacro(ReferenceSeriesInstanceUID);
00064
00065     vtkSetStringMacro(ReferenceFrameOfReferenceUID);
00066     vtkGetStringMacro(ReferenceFrameOfReferenceUID);
00067
00068     // Description:
00069     // Copy the contents of p to this instance.
00070     virtual void DeepCopy(vtkRTStructSetProperties *p);
00071
00072     void AddContourReferencedFrameOfReference( vtkIdType pdnum, const char *classuid , const char * instanceuid );
00073     const char *GetContourReferencedFrameOfReferenceClassUID( vtkIdType pdnum, vtkIdType id );
00074     const char *GetContourReferencedFrameOfReferenceInstanceUID( vtkIdType pdnum, vtkIdType id );
00075     vtkIdType GetNumberOfContourReferencedFrameOfReferences();
00076     vtkIdType GetNumberOfContourReferencedFrameOfReferences(vtkIdType pdnum);
00077
00078     void AddReferencedFrameOfReference( const char *classuid , const char * instanceuid );
00079     const char *GetReferencedFrameOfReferenceClassUID( vtkIdType id );
00080     const char *GetReferencedFrameOfReferenceInstanceUID( vtkIdType id );
00081     vtkIdType GetNumberOfReferencedFrameOfReferences();
00082
00083     void AddStructureSetROI( int roinumber,
00084         const char* refframerefid,
00085         const char* roiname,
00086         const char* ROIGenerationAlgorithm,
00087         const char* ROIDescription = 0
00088     );
00089     void AddStructureSetROIObservation( int refnumber,
00090         int observationnumber,

```


Classes

- class [gdcm::PythonFilter](#)

[PythonFilter](#) [PythonFilter](#) is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace [gdcm](#)

13.640 gdcmPythonFilter.h

[Go to the documentation of this file.](#)

```

00001  /*=====
00002  Program: GDCM (Grassroots DICOM). A DICOM library
00003  Copyright (c) 2006-2011 Mathieu Malaterre
00004  All rights reserved.
00005  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00006
00007  This software is distributed WITHOUT ANY WARRANTY; without even
00008  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00009  PURPOSE. See the above copyright notice for more information.
00010
00011  =====*/
00012
00013
00014 #ifndef GDCMPYTHONFILTER_H
00015 #define GDCMPYTHONFILTER_H
00016
00017 #include <Python.h>
00018
00019 #include "gdcmDataElement.h"
00020 #include "gdcmDicts.h"
00021 #include "gdcmFile.h"
00022
00023 namespace gdcm
00024 {
00025
00026     class GDCM_EXPORT PythonFilter
00027     {
00028     public:
00029         PythonFilter();
00030         ~PythonFilter();
00031
00032         void UseDictAlways(bool ) {}
00033
00034         // Allow user to pass in there own dicts
00035         void SetDicts(const Dicts &dicts);
00036
00037         // Convert to string the ByteValue contained in a DataElement
00038         PyObject *ToPyObject(const Tag& t) const;
00039
00040         void SetFile(const File& f);
00041         File &GetFile();
00042         const File &GetFile() const;
00043
00044     private:
00045         SmartPointer<File> F;
00046     };
00047 } // end namespace gdcm
00048
00049 #endif //GDCMPYTHONFILTER_H

```


Chapter 14

Examples

14.1 TestByteSwap.cxx

This is a C++ example on how to use [gdcm::ByteSwap](#)

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"

#include <cstring> // memcpy

int myfunc()
{
    char vl_str[4];
    const char raw[] = "\000\000\000\004";
    memcpy(vl_str, raw, 4);
    uint32_t vl;
    memcpy(&vl, vl_str, 4);
    gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem(&vl, gdcm::SwapCode::BigEndian, 1);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::LittleEndian);
    if( vl != 0x00000004 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::BigEndian);
    if( vl != 0x40000000 )
    {
        std::cerr << std::hex << "vl: " << vl << std::endl;
        return 1;
    }

    return 0;
}
```

```

}

int TestByteSwap(int , char *[])
{
    gdcmm::SwapCode sc = gdcmm::SwapCode::Unknown;
    if ( gdcmm::ByteSwap<uint16_t>::SystemIsBigEndian() )
    {
        sc = gdcmm::SwapCode::BigEndian;
    }
    else if ( gdcmm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
    {
        sc = gdcmm::SwapCode::LittleEndian;
    }
    if( sc == gdcmm::SwapCode::Unknown )
    {
        std::cerr << "unk" << std::endl;
        return 1;
    }

    //std::cout << "sc: " << sc << std::endl;

    uint16_t t = 0x1234;
    gdcmm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(t, sc);
    if( sc == gdcmm::SwapCode::BigEndian )
    {
        if( t != 0x3412 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
        // ok test pass rest value to old one
        t = 0x1234;
    }
    else if ( sc == gdcmm::SwapCode::LittleEndian )
    {
        if( t != 0x1234 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
    }
}

union { char n[2]; uint16_t tn; } u16;
memcpy(u16.n, &t, 2 );
gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, sc, 1);
uint16_t tn = u16.tn;
if( sc == gdcmm::SwapCode::BigEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
    // ok test pass rest value to old one
    t = 0x1234;
}
else if ( sc == gdcmm::SwapCode::LittleEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, gdcmm::SwapCode::BigEndian, 1);
tn = u16.tn;
if( sc == gdcmm::SwapCode::LittleEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
else if ( sc == gdcmm::SwapCode::BigEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}

```

```

    }

    if( myfunc() )
    {
        return 1;
    }

    uint16_t array[] = { 0x1234 };
    gdcmm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(array,
        gdcmm::SwapCode::BigEndian,1);
    if ( array[0] != 0x3412 )
    {
        std::cerr << std::hex << "array: " << array[0] << std::endl;
        return 1;
    }

    return 0;
}

```

14.2 PatchFile.cxx

This is a C++ example on how to use `gdcmm::Attribute`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 *
 * This C++ code can be used to patch the header.
 */

#include "gdcmmReader.h"
#include "gdcmmImageReader.h"
#include "gdcmmWriter.h"
#include "gdcmmDataSet.h"
#include "gdcmmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *f = argv[1];
    const char *out = argv[2];
    gdcmm::Reader r;
    r.SetFileName( f );
    if( !r.Read() )
    {
        return 1;
    }

    gdcmm::File &file = r.GetFile();
    gdcmm::DataSet& ds = file.GetDataSet();
    // (0028,0100) US 16          # 2, 1 BitsAllocated
    // (0028,0101) US 16          # 2, 1 BitsStored
    // (0028,0102) US 15          # 2, 1 HighBit
    //
    {
        gdcmm::Attribute<0x28,0x100> at;
        at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
        if( at.GetValue() != 8 )
        {

```

```

        return 1;
    }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
        gdcm::Attribute<0x28,0x101> at;
        at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
        if( at.GetValue() != 8 )
        {
            return 1;
        }
        at.SetValue( 32 );
        ds.Replace( at.GetAsDataElement() );
    }
    {
        gdcm::Attribute<0x28,0x102> at;
        at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
        if( at.GetValue() != 7 )
        {
            return 1;
        }
        at.SetValue( 31 );
        ds.Replace( at.GetAsDataElement() );
    }
} // (0028,0008) IS [56] # 2, 1 NumberOfFrames

{
    gdcm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
}

gdcm::Writer w;
w.SetFile( file );
w.SetCheckFileMetaInformation( false );
w.SetFileName( out );
if( !w.Write() )
{
    return 1;
}

// Now let's see if we can read it as an image:
gdcm::ImageReader ir;
ir.SetFileName( out );
if(!ir.Read())
{
    return 1;
}
gdcm::Image &image = ir.GetImage();
unsigned long len = image.GetBufferLength();
const gdcm::ByteValue *bv = ir.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) ).GetByteValue();
if( !bv || len != bv->GetLength() )
{
    return 1;
}
std::cout << bv->GetLength() << " " << len << std::endl;

std::cout << "Success to rewrite image !" << std::endl;
image.Print( std::cout );
return 0;
}

```

14.3 SimplePrint.cs

This is a C# example on how to use gdcm::SWIGDataSet

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
/*
  Convertor convertor = new Convertor();
  int a = convertor.Convert<int>( some_int_blob );
  double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
  public static void RecurseDataSet(File f, DataSet ds, string indent)
  {
    CSharpDataSet cds = new CSharpDataSet(ds);
    while(!cds.IsAtEnd())
    {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
      {
        uint uvl = (uint)de.GetVL(); // Test cast is ok
        System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
        SmartPtrSQ sq = de.GetValueAsSQ();
        uint n = sq.GetNumberOfItems();
        for( uint i = 1; i <= n; i++) // item starts at 1, not 0
        {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + " " );
        }
      }
      else
      {
        System.Console.WriteLine( indent + de.toString() );
      }
      cds.Next();
    }
  }

  public static int Main(string[] args)
  {
    string filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
    {
      return 1;
    }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    RecurseDataSet( f, ds, "" );

    return 0;
  }
}

```

14.4 TestReader.cxx

```

/*=====
  Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmFile.h"
#include "gdcmTesting.h"
#include "gdcmMediaStorage.h"

int TestRead(const char* filename, bool verbose = false)
{
    if( verbose )
        std::cout << "TestRead: " << filename << std::endl;

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if ( !reader.Read() )
    {
        std::cerr << "TestReadError: Failed to read: " << filename << std::endl;
        return 1;
    }

    //commenting out the fmi and ds to avoid warnings
    //const gdcm::FileMetaInformation &h = reader.GetFile().GetHeader();
    //std::cout << h << std::endl;

    //const gdcm::DataSet &ds = reader.GetFile().GetDataSet();
    //std::cout << ds << std::endl;

    const char *ref = gdcm::Testing::GetMediaStorageFromFile(filename);
    gdcm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( !ref )
    {
        std::cerr << "TestReadError: Missing MediaStorage: " << filename << std::endl;
        std::cerr << "It should be: " << ms << std::endl;
        return 1;
    }

    if( ms.IsUndefined() && ref && *ref != 0 )
    {
        std::cerr << "TestReadError: MediaStorage: " << filename << std::endl;
        std::cerr << "It should be instead: " << ref << std::endl;
        return 1;
    }

    // Make sure it is the right one:

    if( ref && *ref != 0 && ms != gdcm::MediaStorage::GetMSType(ref) )
    {
        std::cerr << "Error: Found MediaStorage: " << ms << " for " << filename << std::endl;
        std::cerr << "It should be instead: " << ref << std::endl;
        return 1;
    }

    return 0;
}

int TestReader(int argc, char *argv[])
{
    if( argc == 2 )
    {
        const char *filename = argv[1];
        return TestRead(filename, true);
    }

    // else
    gdcm::Trace::DebugOff();
    gdcm::Trace::WarningOff();
    int r = 0, i = 0;
    const char *filename;
    const char * const *filenames = gdcm::Testing::GetFileNames();
    while( (filename = filenames[i]) )
    {

```



```

    r += TestRead( filename );
    ++i;
}

return r;
}

```

14.5 TestReader.py

This is a C++ example on how to use [gdcm::Reader](#)

```

00001
00014
00015 import os,sys
00016 import gdcm
00017
00018 def TestRead(filename, verbose = False):
00019     r = gdcm.Reader()
00020     r.SetFileName( filename )
00021     success = r.Read()
00022     #if verbose: print r.GetFile()
00023     if verbose: print(r.GetFile().GetDataSet())
00024     return success
00025
00026 if __name__ == "__main__":
00027     success = 0
00028     try:
00029         filename = os.sys.argv[1]
00030         success += TestRead( filename, True )
00031     except:
00032         # loop over all files:
00033         gdcm.Trace.DebugOff()
00034         gdcm.Trace.WarningOff()
00035         t = gdcm.Testing()
00036         nfiles = t.GetNumberOfFileNames()
00037         for i in range(0,nfiles):
00038             filename = t.GetFileName(i)
00039             success += TestRead( filename )
00040
00041
00042 # Test succeed ?
00043 sys.exit(success == 0)

```

14.6 DecompressJPEGFile.cs

This is a C# example on how to use [gdcm::SequenceOfFragments](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile

```

```

{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        System.IO.FileStream infile =
            new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
        uint fsize = gdcm.PosixEmulation.FileSize(file1);

        byte[] jstream = new byte[fsize];
        infile.Read(jstream, 0, jstream.Length);

        Trace.DebugOn();
        Image image = new Image();
        image.SetNumberOfDimensions( 2 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated syntax
        // in which can one cannot use a simple byte array for storage. Instead, see
        // gdcm.SequenceOfFragments
        // pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
        SmartPtrFrag sq = SequenceOfFragments.New();
        Fragment frag = new Fragment();
        frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );
        // Single file => single fragment
        sq.AddFragment( frag );
        // Pass by reference:
        pixeldata.SetValue( sq.____ref____() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
        // FIXME hardcoded:
        PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_FULL );
        image.SetPhotometricInterpretation( pi );
        // FIXME hardcoded:
        PixelFormat pixeltype = new PixelFormat(3,8,8,7);
        image.SetPixelFormat( pixeltype );

        // FIXME hardcoded:
        image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
        image.SetDimension(0, 692);
        image.SetDimension(1, 721);

        // Decompress !
        byte[] decompressedData = new byte[(int)image.GetBufferLength()];
        image.GetBuffer(decompressedData);

        // Write out the decompressed bytes
        System.Console.WriteLine(image.toString());
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/dd.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(decompressedData);
        }

        return 0;
    }
}

```

14.7 ManipulateFile.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ManipulateFile.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class ManipulateFile
{
    public static int Main(string[] args)
    {
        {
            string file1 = args[0];
            string file2 = args[1];
            Reader reader = new Reader();
            reader.SetFileName( file1 );
            bool ret = reader.Read();
            if( !ret )
            {
                return 1;
            }

            Anonymizer ano = new Anonymizer();
            ano.SetFile( reader.GetFile() );
            ano.RemovePrivateTags();
            ano.RemoveGroupLength();
            Tag t = new Tag(0x10,0x10);
            ano.Replace( t, "GDCM^Csharp^Test^Hello^World" );

            UIDGenerator g = new UIDGenerator();
            ano.Replace( new Tag(0x0008,0x0018), g.Generate() );
            ano.Replace( new Tag(0x0020,0x000d), g.Generate() );
            ano.Replace( new Tag(0x0020,0x000e), g.Generate() );
            ano.Replace( new Tag(0x0020,0x0052), g.Generate() );

            Writer writer = new Writer();
            writer.SetFileName( file2 );
            writer.SetFile( ano.GetFile() );
            ret = writer.Write();
            if( !ret )
            {
                return 1;
            }

            return 0;
        }
    }
}

```

14.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use Anonymizer

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Typical usage on UNIX:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
 */
using System;

```

```

using gdcM;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void StartFilter() {
        System.Console.WriteLine( "This is my start" );
    }
    protected override void EndFilter(){
        System.Console.WriteLine( "This is my end" );
    }
    protected override void ShowProgress(Subject caller, Event evt){
        ProgressEvent pe = ProgressEvent.Cast(evt);
        System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
    }
    protected override void ShowIteration(){
        System.Console.WriteLine( "This is my iteration" );
    }
    protected override void ShowAnonymization(Subject caller, Event evt){
        /*
        * A couple of explanation are necessary here to understand how SWIG work
        * http://www.swig.org/Doc1.3/Java.html#adding\_downcasts
        *
        * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
        * System.Type type = evt.GetType();
        * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
        * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
        * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
        */
        AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
        if( ae != null )
        {
            Tag t = ae.GetTag();
            System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
    protected override void ShowAbort(){
        System.Console.WriteLine( "This is my abort" );
    }
}

public class ClinicalTrialIdentificationWorkflow
{
    public static bool ProcessOneFile( gdcM.Anonymizer ano , string filename, string outfilename )
    {
        Reader reader = new Reader();
        reader.SetFileName( filename );
        bool ret = reader.Read();
        if( !ret )
        {
            return false;
        }
        // Pass in the file:
        ano.SetFile( reader.GetFile() );

        // First step, let's protect all Patient information as per
        // PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
        if( !ano.BasicApplicationLevelConfidentialityProfile() )
        {
            return false;
        }

        // Now let's pass in all Clinical Trial fields
        // PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
        /*
        Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial sponsor. See C.7.1.3.1.1.
        Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol. See C.7.1.3.1.2.
        Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial protocol. See C.7.1.3.1.3.
        Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible for submitting clinical trial data. See C.7.1.3.1.4.
        Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for submitting clinical trial data. See C.7.1.3.1.5
        Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the clinical trial subject. See C.7.1.3.1.6. Shall be present if
        Clinical Trial Subject Reading ID (0012,0042) is absent. May be present otherwise.
        Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for blinded evaluations. Shall be present if Clinical Trial
        Subject ID (0012,0040) is absent. May be present otherwise. See C.7.1.3.1.7.
        */
        ano.Replace( new gdcM.Tag(0x0012,0x0010), "MySponsorName");
        ano.Replace( new gdcM.Tag(0x0012,0x0020), "MyProtocolID");
    }
}

```

```

ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

// The following two are not required as they are guaranteed to be filled in by the
// Basic Application Level Confidentiality Profile. Only override if you understand what
// you are doing
//ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
//ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");

// We might be generating a subdirectory. Let's make sure the subdir exist:
gdcm.Filename fn = new gdcm.Filename( outfilename );
string subdir = fn.GetPath();
if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
{
    return false;
}

gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

Writer writer = new Writer();
writer.SetFileName( outfilename );
writer.SetFile( ano.GetFile() );
ret = writer.Write();
if( !ret )
{
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
    {
        System.Console.WriteLine( "Could not LoadResourcesFiles" );
        return 1;
    }

    if( args.Length != 2 )
    {
        System.Console.WriteLine( "Usage:" );
        System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir output_dir" );
        return 1;
    }
    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    // Recursively search all file within this toplevel directory:
    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );

```

```

if(nfiles == 0) return 1;

// Let's use the pre-shipped certificate of GDCM.
string certpath = gdcml.Filename.Join(gdcml.Testing.GetSourceDirectory(), "/Testing/Source/Data/certificate.pem" );
gdcml.CryptoFactory fact = gdcml.CryptoFactory.GetFactoryInstance();
gdcml.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
if( !cms.ParseCertificateFile( certpath ) )
{
    System.Console.WriteLine( "PEM Certificate : " + certpath + " could not be read. Sorry" );
    return 1;
}

//Anonymizer ano = new Anonymizer();
// A reference to an actual C++ instance is required here:
SmartPtrAno sano = Anonymizer.New();
Anonymizer ano = sano.__ref__();

//SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
MyWatcher watcher = new MyWatcher(ano);

// Explicitly specify the Cryptographic Message Syntax to use:
ano.SetCryptographicMessageSyntax( cms );

// Process all filenames:
FilenamesType filenames = d.GetFilesNames();
for( uint i = 0; i < nfiles; ++i )
{
    string filename = filenames[ (int)i ];
    string outfilename = filename.Replace( dir1, dir2 );
    System.Console.WriteLine( "Filename: " + filename );
    System.Console.WriteLine( "Out Filename: " + outfilename );
    if( !ProcessOneFile( ano , filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
    }
}

return 0;
}
}

```

14.9 GenerateDICOMDIR.cs

This is a C# example on how to use DICOMDIRGenerator

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcm;

public class GenerateDICOMDIR
{
    public static int Main(string[] args)
    {
        string directory = args[0];
    }
}

```

```

string outfilename = args[1];

Directory d = new Directory();
uint nfiles = d.Load( directory, true );
if(nfiles == 0) return 1;
//System.Console.WriteLine( "Files:\n" + d.toString() );

// Implement fast path ?
// Scanner s = new Scanner();

string descriptor = "My_Descriptor";
FileNamesType filenames = d.GetFilesNames();

gdcmm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
gen.SetFileNames( filenames );
gen.SetDescriptor( descriptor );
if( !gen.Generate() )
{
    return 1;
}

gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
gdcmm.Writer writer = new Writer();
writer.SetFile( gen.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}
}

```

14.10 GenFakeImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmImage.h"
#include "gdcmmImageWriter.h"
#include "gdcmmFileDerivation.h"
#include "gdcmmUIDGenerator.h"
// #include "gdcmmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcmm.FileDerivation filter. This filter is meant to create "DERIVED" image
 * object. FileDerivation has a simple API where you can reference *all* the input image that have been
 * used to generate the image. The API also allows user to specify the purpose of reference (see CID 7202,
 * PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
 */
int main(int, char *[])
{
    // Step 1: Fake Image
    gdcmm::SmartPointer<gdcmm::Image> im = new gdcmm::Image;

    char * buffer = new char[ 256 * 256 * 3];
    char * p = buffer;
    int b = 128;
    //int ybr[3];
    int ybr2[3];
    //int rgb[3];

    for(int r = 0; r < 256; ++r)
        for(int g = 0; g < 256; ++g)

```

```

        //for(int b = 0; b < 256; ++b)
        {
            //rgb[0] = r;
            //rgb[1] = g;
            //rgb[1] = 128;
            //rgb[2] = b;
            //ybr[0] = r;
            //ybr[1] = g;
            //ybr[1] = 128;
            //ybr[2] = b;

            ybr2[0] = r;
            ybr2[1] = g;
            ybr2[1] = 128;
            ybr2[2] = b;
            //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
            //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
            *p++ = (char)ybr2[0];
            *p++ = (char)ybr2[1];
            *p++ = (char)ybr2[2];
        }

im->SetNumberOfDimensions( 2 );
im->SetDimension(0, 256 );
im->SetDimension(1, 256 );

im->GetPixelFormat().SetSamplesPerPixel(3);
//im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::YBR_FULL );

unsigned long l = im->GetBufferLength();
if( l != 256 * 256 * 3 )
{
    return 1;
}
gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buffer, (uint32_t)l );
delete[] buffer;
im->SetDataElement( pixeldata );

gdcm::UIDGenerator uid; // helper for uid generation

gdcm::SmartPointer<gdcm::File> file = new gdcm::File; // empty file

// Step 2: DERIVED object
gdcm::FileDerivation fd;
// For the purpose of this exercise we will pretend that this image is referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// { "DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 113072 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
{
    std::cerr << "Sorry could not derive using input info" << std::endl;
    return 1;
}

// We pass both :
// 1. the fake generated image
// 2. the 'DERIVED' dataset object
// to the writer.
gdcm::ImageWriter w;
w.SetImage( *im );
w.SetFile( fd.GetFile() );

// Set the filename:
w.SetFileName( "ybr2.dcm" );
if( !w.Write() )
{
    return 1;
}

```



```

    }
    return 0;
}

```

14.11 ReformatFile.cs

This is a C++ example on how to use FileDerivation

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ReformatFile.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class ReformatFile
{
    public static int Main(string[] args)
    {
        gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App" );

        // http://www.oid-info.com/get/1.3.6.1.4.17434
        string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
        gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
        System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

        string filename = args[0];
        string outfilename = args[1];

        Reader reader = new Reader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
            return 1;
        }

        UIDGenerator uid = new UIDGenerator(); // helper for uid generation
        FileDerivation fd = new FileDerivation();
        // For the purpose of this exercise we will pretend that this image is referencing
        // two source image (we need to generate fake UID for that).
        string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
        fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

        // Again for the purpose of the exercise we will pretend that the image is a
        // multiplanar reformat (MPR):
        // CID 7202 Source Image Purposes of Reference
        // { "DCM",121322,"Source image for image processing operation"},
        fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
        // CID 7203 Image Derivation
        // { "DCM",113072,"Multiplanar reformatting" },
        fd.SetDerivationCodeSequenceCodeValue( 113072 );
        fd.SetFile( reader.GetFile() );
        // If all Code Value are ok the filter will execute properly
        if( !fd.Derive() )
        {
            return 1;
        }
    }
}

```

```

    }

    gdcM.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcM.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcM.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcM.Tag(0x0002,0x0016) );

    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( fd.GetFile() );
    if( !writer.Write() )
    {
        System.Console.WriteLine( "Could not write: " + outfilename );
        return 1;
    }

    return 0;
}

```

14.12 DecompressImage.cs

This is a C# example on how to use Image

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcM/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcMData/012345.002.050.dcm decompress.dcm
 */
using System;
using gdcM;

public class DecompressImage
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        string file2 = args[1];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        // check that one can access a Fragment from C#:
        var de = reader.GetFile().GetDataSet().GetDataElement(new Tag(0x7fe0, 0x0010));
        var sq = de.GetSequenceOfFragments();
        sq.GetFragment(0);

        Image image = new Image();
        Image ir = reader.GetImage();

        image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );

        //Just for fun:
        //int dircos = ir.GetDirectionCosines();
        //t = gdcM.Orientation.GetType(dircos);
        //int l = gdcM.Orientation.GetLabel(t);

```

```

//System.Console.WriteLine( "Orientation label:" + l );

// Set the dimensions,
// 1. either one at a time
//image.SetDimension(0, ir.GetDimension(0) );
//image.SetDimension(1, ir.GetDimension(1) );

// 2. the array at once
uint[] dims = {0, 0};
// Just for fun let's invert the dimensions:
dims[0] = ir.GetDimension(1);
dims[1] = ir.GetDimension(0);
ir.SetDimensions( dims );

PixelFormat pixeltype = ir.GetPixelFormat();
image.SetPixelFormat( pixeltype );

PhotometricInterpretation pi = ir.GetPhotometricInterpretation();
image.SetPhotometricInterpretation( pi );

DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
byte[] str1 = new byte[ ir.GetBufferLength()];
ir.GetBuffer( str1 );
//System.Console.WriteLine( ir.GetBufferLength() );
pixeldata.SetByteValue( str1, new VL( (uint)str1.Length ) );
//image.SetDataElement( pixeldata );
ir.SetDataElement( pixeldata );

ImageWriter writer = new ImageWriter();
writer.SetFileName( file2 );
writer.SetFile( reader.GetFile() );
writer.SetImage( ir );
ret = writer.Write();
if ( !ret )
{
    return 1;
}

return 0;
}
}

```

14.13 StandardizeFiles.cs

This is a C++ example on how to use ImageChangeTransferSyntax

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
 */
using System;
using gdcm;

public class StandardizeFiles
{
    public static bool ProcessOneFile( string filename, string outfilename )
    {
        PixmapReader reader = new PixmapReader();
    }
}

```

```

reader.SetFileName( filename );
if( !reader.Read() )
{
    System.Console.WriteLine( "Could not read: " + filename );
    return false;
}

ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
change.SetForce( false ); // do we really want to recompress when input is already compressed in same alg ?
change.SetCompressIconImage( false ); // Keep it simple
change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEG2000Lossless ) );
change.SetInput( reader.GetPixmap() );
if( !change.Change() )
{
    System.Console.WriteLine( "Could not change: " + filename );
    return false;
}

gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

PixmapWriter writer = new PixmapWriter();
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
gdcm.Pixmap pixout = ((PixmapToPixmapFilter)change).GetOutput();

writer.SetPixmap( pixout );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write: " + outfilename );
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory( dir1 ) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory( dir2 ) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
    {
        string filename = filenames[ (int)i ];
        string outfilename = filename.Replace( dir1, dir2 );
        System.Console.WriteLine( "Filename: " + filename );
        System.Console.WriteLine( "Out Filename: " + outfilename );
        if( !ProcessOneFile( filename, outfilename ) )
        {
            System.Console.WriteLine( "Could not process filename: " + filename );
            //return 1;
        }
    }
}

```

```

    }

    return 0;
}
}

```

14.14 ScanDirectory.cs

This is a C# example on how to use Scanner

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

// We will print each filename being processed
public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void ShowFileName(Subject caller, Event evt){
        FileNameEvent fne = FileNameEvent.Cast(evt);
        if( fne != null )
        {
            string fn = fne.GetFileName();
            System.Console.WriteLine( "This is my Scanner. Processing FileName: " + fn );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
}

public class ScanDirectory
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        Tag t = new Tag(0x8,0x80);

        Directory d = new Directory();
        uint nfiles = d.Load( directory );
        if(nfiles == 0) return 1;
        //System.Console.WriteLine( "Files:\n" + d.toString() );

        // Use a StrictScanner, need to use a reference to pass the C++ pointer to
        // MyWatcher implementation
        SmartPtrStrictScan sscan = StrictScanner.New();
        StrictScanner s = sscan.__ref__();
        MyWatcher watcher = new MyWatcher(s);

        s.AddTag( t );
        bool b = s.Scan( d.GetFilenames() );
        if(!b) return 1;

        for(int i = 0; i < (int)nfiles; ++i)
        {
            if( !s.IsKey( d.GetFilenames()[i] ) )
            {

```

```

        System.Console.WriteLine( "File is not DICOM or could not be read: " + d.GetFileNames()[i] );
    }
}

System.Console.WriteLine( "Scan:\n" + s.toString() );

System.Console.WriteLine( "success" );
return 0;
}
}

```

14.15 BasicAnonymizer.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/BasicAnonymizer.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
        protected override void StartFilter() {
            System.Console.WriteLine( "This is my start" );
        }
        protected override void EndFilter(){
            System.Console.WriteLine( "This is my end" );
        }
        protected override void ShowProgress(Subject caller, Event evt){
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
        }
        protected override void ShowIteration(){
            System.Console.WriteLine( "This is my iteration" );
        }
        protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
            AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
            if( ae != null )
            {
                Tag t = ae.GetTag();
                System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
            }
            else
            {
                System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
            }
        }
        protected override void ShowAbort(){
            System.Console.WriteLine( "This is my abort" );
        }
    }
}

```

```

}

public class BasicAnonymizer
{
    public static int Main(string[] args)
    {
        gdcm.Global global = gdcm.Global.GetInstance();
        if( !global.LoadResourcesFiles() )
        {
            System.Console.WriteLine( "Could not LoadResourcesFiles" );
            return 1;
        }

        string file1 = args[0];
        string file2 = args[1];
        Reader reader = new Reader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(), "/Testing/Source/Data/certificate.pem" );
        gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
        gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
        if( !cms.ParseCertificateFile( certpath ) )
        {
            return 1;
        }

        //Anonymizer ano = new Anonymizer();
        SmartPtrAno sano = Anonymizer.New();
        Anonymizer ano = sano.__ref__();

        //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
        MyWatcher watcher = new MyWatcher(ano);

        ano.SetFile( reader.GetFile() );
        ano.SetCryptographicMessageSyntax( cms );
        if( !ano.BasicApplicationLevelConfidentialityProfile() )
        {
            return 1;
        }

        Writer writer = new Writer();
        writer.SetFileName( file2 );
        writer.SetFile( ano.GetFile() );
        ret = writer.Write();
        if( !ret )
        {
            return 1;
        }

        return 0;
    }
}

```

14.16 BasicImageAnonymizer.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
*/
using System;

```

```

using gdcm;

public class BasicImageAnonymizer
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageReader reader = new gdcm.ImageReader();
        reader.SetFileName( filename );

        if (!reader.Read()) return 1;

        Image ir = reader.GetImage();

        uint[] dims = {0, 0, 0};
        dims[0] = ir.GetDimension(0);
        dims[1] = ir.GetDimension(1);
        dims[2] = ir.GetDimension(2);
        System.Console.WriteLine( "Dim:" + dims[0] );
        System.Console.WriteLine( "Dim:" + dims[1] );
        System.Console.WriteLine( "Dim:" + dims[2] );

        // buffer to get the pixels
        byte[] buffer = new byte[ ir.GetBufferLength() ];
        System.Console.WriteLine( "Dim:" + ir.GetBufferLength() );
        ir.GetBuffer( buffer );

        for (uint z = 0; z < dims[2]; z++)
        {
            for (uint y = 0; y < dims[1] / 2; y++) // only half Y
            {
                for (uint x = 0; x < dims[0] / 2; x++) // only half X
                {
                    buffer[ (z * dims[1] + y) * dims[0] + x ] = 0; // works when pixel type == UINT8
                }
            }
        }

        DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
        pixeldata.SetByteValue( buffer, new VL( (uint)buffer.Length ) );
        ir.SetDataElement( pixeldata );
        ir.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.ExplicitVRLittleEndian ) );

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLSLossless ) );
        change.SetInput( ir );
        if( !change.Change() )
        {
            System.Console.WriteLine( "Could not change: " + filename );
            return 1;
        }

        ImageWriter writer = new ImageWriter();
        writer.SetFileName( "out.dcm" );
        writer.SetFile( reader.GetFile() );
        writer.SetImage( change.GetOutput() );
        bool ret = writer.Write();
        if( !ret )
        {
            return 1;
        }

        return 0;
    }
}

```

14.17 Cleaner.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```


See Copyright.txt or <http://gdcms.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/Cleaner.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void StartFilter() {
        System.Console.WriteLine( "This is my start" );
    }
    protected override void EndFilter(){
        System.Console.WriteLine( "This is my end" );
    }
    protected override void ShowProgress(Subject caller, Event evt){
        ProgressEvent pe = ProgressEvent.Cast(evt);
        System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
    }
    protected override void ShowIteration(){
        System.Console.WriteLine( "This is my iteration" );
    }
    protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding\_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
        AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
        if( ae != null )
        {
            Tag t = ae.GetTag();
            System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
    protected override void ShowAbort(){
        System.Console.WriteLine( "This is my abort" );
    }
}

public class Cleaner
{
    public static int Main(string[] args)
    {
        gdcm.Global global = gdcm.Global.GetInstance();
        if( !global.LoadResourcesFiles() )
        {
            System.Console.WriteLine( "Could not LoadResourcesFiles" );
            return 1;
        }

        string file1 = args[0];
        string file2 = args[1];
        Reader reader = new Reader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        SmartPtrCleaner scleaner = gdcm.Cleaner.New();

```

```

gdcM.Cleaner cleaner = scleaner.__ref__();

//SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(cleaner, "Anonymizer");
MyWatcher watcher = new MyWatcher(cleaner);

cleaner.SetFile( reader.GetFile() );
cleaner.Empty( new gdcM.VR(gdcM.VR.VRType.PN) );
gdcM.DPath dpath = new gdcM.DPath();
dpath.ConstructFromString( " /0010,0010" );
cleaner.Preserve( dpath );
gdcM.Tag t1 = new gdcM.Tag(0x10, 0x30);
cleaner.Empty( t1 );
gdcM.PrivateTag pt0 = new gdcM.PrivateTag( new gdcM.Tag(0x29,0x60), "SIEMENS MEDCOM HEADER2" );
cleaner.Remove( pt0 );
gdcM.PrivateTag pt1 = new gdcM.PrivateTag( new gdcM.Tag(0x29,0x10), "SIEMENS CSA HEADER" );
gdcM.PrivateTag pt2 = new gdcM.PrivateTag( new gdcM.Tag(0x29,0x20), "SIEMENS CSA HEADER" );
cleaner.Scrub( pt1 );
cleaner.Scrub( pt2 );
if( !cleaner.Clean() )
{
    return 1;
}

Writer writer = new Writer();
writer.SetFileName( file2 );
writer.SetFile( cleaner.GetFile() );
ret = writer.Write();
if( !ret )
{
    return 1;
}

return 0;
}
}

```

14.18 CompressLossyJPEG.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Perso/gdcM/debug-gcc/bin
* $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
*/

using System;
using gdcM;

public class CompressLossyJPEG
{
    public static int Main(string[] args)
    {
        if( args.Length < 2 )
        {
            System.Console.WriteLine( " input.dcm output.dcm" );
            return 1;
        }
        string filename = args[0];
        string outfilename = args[1];

        ImageReader reader = new ImageReader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
        }
    }
}

```

```

    return 1;
}

// The output of gdcm::Reader is a gdcm::File
File file = reader.GetFile();

// the dataset is the the set of element we are interested in:
DataSet ds = file.GetDataSet();

Image image = reader.GetImage();
//image.Print( cout );

ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
TransferSyntax targetts = new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
change.SetTransferSyntax( targetts );

// Setup our JPEGCodec, warning it should be compatible with JPEGBaselineProcess1
JPEGCodec jpegcodec = new JPEGCodec();
if( !jpegcodec.CanCode( targetts ) )
{
    System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot handle JPEGBaselineProcess1" );
    return 1;
}
jpegcodec.SetLossless( false );
jpegcodec.SetQuality( 50 ); // poor quality !
change.SetUserCodec( jpegcodec ); // specify the codec to use to the ImageChangeTransferSyntax

change.SetInput( image );
bool b = change.Change();
if( !b )
{
    System.Console.WriteLine( "Could not change the Transfer Syntax" );
    return 1;
}

ImageWriter writer = new ImageWriter();
writer.SetImage( (gdcm.Image)change.GetOutput() );
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write: " + outfilename );
    return 1;
}

return 0;
}
}

```

14.19 DecompressImageMultiframe.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
$ gdcminfo ~/Desktop/angiogram-06.dcm
MediaStorage is 1.2.840.10008.5.1.4.1.1.12.1 [X-Ray Angiographic Image Storage]
TransferSyntax is 1.2.840.10008.1.2.4.50 [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image
Compression]
NumberOfDimensions: 3
Dimensions: (512,512,355)
Origin: (0,0,0)
Spacing: (1,1,40)
DirectionCosines: (1,0,0,0,1,0)
Rescale Intercept/Slope: (0,1)

```

```

SamplesPerPixel :1
BitsAllocated :8
BitsStored :8
HighBit :7
PixelRepresentation:0
ScalarType found :UINT8
PhotometricInterpretation: MONOCHROME2
PlanarConfiguration: 0
TransferSyntax: 1.2.840.10008.1.2.4.50
Orientation Label: AXIAL
*/

/*
 * Description:
 *
 * Assume we have a file angiogram-06.dcm as described above.
 * the following program will decompress directly from the extracted jpeg stream.
 *
 * First step extract the jpeg stream (but not the Basic Offset Table):
 *
 * $ gdcmmraw -i angiogram-06.dcm -o /tmp/output/chris --split-frags --pattern %d.jpg
 *
 * Check that indeed there are 355 files, while there are 356 fragments in the original DICOM file, since
 * gdcmmraw always skip the first fragment (Basic Offset Table).
 *
 * Now from those individual jpeg stream, recreate a fake gdcm.DataElement...
 *
 * Usage:
 *
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono ./bin/DecompressImageMultiframe.exe /tmp/output
 */
using System;
using gdcm;

public class DecompressImageMultiframe
{
    public static int Main(string[] args)
    {
        string directory = args[0];
        gdcm.Directory dir = new gdcm.Directory();
        uint nfiles = dir.Load(directory);
        //System.Console.WriteLine(dir.ToString());
        gdcm.FilenamesType filenames = dir.GetFilenames();

        Image image = new Image();
        image.SetNumberOfDimensions( 3 ); // important for now
        DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

        // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
        SmartPtrFrag sq = SequenceOfFragments.New();

        // Yeah, the files are not guaranteed to be in order, please adapt...
        for(uint i = 0; i < nfiles; ++i)
        {
            System.Console.WriteLine( filenames[(int)i] );
            string file = filenames[(int)i];
            System.IO.FileStream infile =
                new System.IO.FileStream(file, System.IO.FileMode.Open, System.IO.FileAccess.Read);
            uint fsize = gdcm.PosixEmulation.FileSize(file);

            byte[] jstream = new byte[fsize];
            infile.Read(jstream, 0 , jstream.Length);

            Fragment frag = new Fragment();
            frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
            sq.AddFragment( frag );
        }

        // Pass by reference:
        pixeldata.SetValue( sq.__ref__() );

        // insert:
        image.SetDataElement( pixeldata );

        // JPEG use YBR to achieve better compression ratio by default (not RGB)
        // FIXME hardcoded:
        PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2 );
        image.SetPhotometricInterpretation( pi );
        // FIXME hardcoded:
        PixelFormat pixeltype = new PixelFormat(1,8,8,7);
    }
}

```

```

image.SetPixelFormat( pixeltype );

// FIXME hardcoded:
image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
image.SetDimension(0, 512);
image.SetDimension(1, 512);
image.SetDimension(2, 355);

// Decompress !
byte[] decompressedData = new byte[(int)image.GetBufferLength()];
image.GetBuffer(decompressedData);

// Write out the decompressed bytes
System.Console.WriteLine(image.toString());
using (System.IO.Stream stream =
    System.IO.File.Open(@"tmp/dd.raw",
        System.IO.FileMode.Create))
{
    System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
    writer.Write(decompressedData);
}

return 0;
}
}

```

14.20 DumpCSA.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ bin/DumpCSA.exe input.dcm
 */
using System;
using gdcm;

public class DumpCSA
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        gdcm.Reader reader = new gdcm.Reader();
        reader.SetFileName( filename );
        if (!reader.Read()) return 1;

        gdcm.File f = reader.GetFile();
        gdcm.DataSet ds = f.GetDataSet();

        string[] expectedSiemensTags = new string[] { "B_value", "AcquisitionMatrixText" };
        using (PrivateTag gtag = CSAHeader.GetCSAImageHeaderInfoTag())
        {
            if (ds.FindDataElement(gtag))
            {
                using (DataElement de = ds.GetDataElement(gtag))
                {
                    if (de != null && !de.IsEmpty())
                    {
                        using (CSAHeader csa = new CSAHeader())
                        {
                            if (csa.LoadFromDataElement(de))
                            {
                                foreach (string str in expectedSiemensTags)

```

```

        {
            if (csa.FindCSAElementByName(str))
            {
                using (CSAElement elem = csa.GetCSAElementByName(str))
                {
                    if (elem != null)
                    {
                        System.Console.WriteLine( elem.ToString() );
                    }
                }
            }
        }
    }
}

return 0;
}
}

```

14.21 ExplicitLittleEndian.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
 */
using System;
using gdcm;

public class ExplicitLittleEndian
{
    public static bool ToExplicitLittleEndian(string aSrc, string aDst)
    {
        using (ImageReader reader = new ImageReader())
        {
            bool b = System.IO.File.Exists(aSrc);
            reader.SetFileName(aSrc);
            if (!reader.Read())
            {
                throw new System.Exception(string.Format("Cannot read '{0}'", aSrc));
            }
            using (FileExplicitFilter fef = new FileExplicitFilter())
            {
                fef.SetChangePrivateTags(false);
                fef.SetFile(reader.GetFile());
                if (!fef.Change())
                {
                    throw new System.Exception(string.Format("Cannot make explicit '{0}'", aSrc));
                }
                using (var syntax = new TransferSyntax(TransferSyntax.TSType.ExplicitVRLittleEndian))
                {
                    using (ImageChangeTransferSyntax tsc = new ImageChangeTransferSyntax())
                    {
                        tsc.SetTransferSyntax(syntax);
                        tsc.SetInput(reader.GetImage());
                        tsc.SetForce(true);
                    }
                }
            }
        }
    }
}

```

```

        if (!tsc.Change())
        {
            throw new System.Exception(string.Format("Cannot change '{0}'", aSrc));
        }
        using (var writer = new ImageWriter())
        {
            writer.SetFile(fef.GetFile());
            writer.SetImage(tsc.GetOutput());
            writer.SetFileName(aDst);
            if (!writer.Write())
            {
                throw new System.Exception(string.Format("Cannot write to '{0}'", aDst));
            }
        }
    }
}
}
}
return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
    {
        string filename = filenames[ (int)i ];
        string outfilename = filename.Replace( dir1, dir2 );
        System.Console.WriteLine( "Filename: " + filename );
        System.Console.WriteLine( "Out Filename: " + outfilename );
        if( !ToExplicitLittleEndian( filename, outfilename ) )
        {
            System.Console.WriteLine( "Could not process filename: " + filename );
            //return 1;
        }
    }

    return 0;
}
}

```

14.22 ExtractEncapsulatedFile.cs

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * This example shows how one from C# context can extract a binary blob
 * and write out as a file.
 * This example is meant for pdf encapsulated file, but can be adapted for other type
 * of binary blob.
 *
 * DICOM file is:
 * ...
 * (0042,0010) ST (no value available) # 0, 0 DocumentTitle
 * (0042,0011) OB 25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1 EncapsulatedDocument
 * (0042,0012) LO [application/pdf] # 16, 1 MIMETimeTypeOfEncapsulatedDocument
 * ...
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
 */
using System;
using gdcm;

public class ExtractEncapsulatedFile
{
    public static int Main(string[] args)
    {
        string file = args[0];
        Reader reader = new Reader();
        reader.SetFileName( file );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();
        Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
        if( !ds.FindElement( tencapsulated_stream ) )
        {
            return 1;
        }
        // else
        DataElement de = ds.GetDataElement( tencapsulated_stream );
        ByteValue bv = de.GetByteValue();
        uint len = bv.GetLength();
        byte[] encapsulated_stream = new byte[len];
        bv.GetBuffer( encapsulated_stream, len );

        // Write out the decompressed bytes
        //System.Console.WriteLine(image.toString());
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/dd.pdf",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write( encapsulated_stream );
        }

        return 0;
    }
}

```

14.23 ExtractImageRegion.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```


All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ bin/ExtractImageRegion.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegion.exe gdcmData/012345.002.050.dcm
 * $ md5sum /tmp/frame.raw
 * d594a5e2fde12f32b6633ca859b4d4a6 /tmp/frame.raw
 * $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
 * [...]
 * md5sum: d594a5e2fde12f32b6633ca859b4d4a6
 */
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        uint file_size = gdcm.PosixEmulation.FileSize(filename);

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return 1;

        // store current offset:
        uint cur_pos = reader.GetStreamCurrentPosition();

        uint remaining = file_size - cur_pos;

        Console.WriteLine("Remaining bytes to read (Pixel Data): " + remaining.ToString() );

        // Get file infos
        gdcm.File f = reader.GetFile();

        // get some info about image
        UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
        PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
        int pixelsize = pf.GetPixelSize();
        PhotometricInterpretation pi = ImageHelper.GetPhotometricInterpretationValue(f);
        Console.WriteLine( pi.ToString() );

        // buffer to get the pixels
        byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

        // define a simple box region.
        BoxRegion box = new BoxRegion();
        for (uint z = 0; z < dims[2]; z++)
        {
            // Define that I want the image 0, full size (dimx x dimy pixels)
            // and do that for each z:
            box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
            //System.Console.WriteLine( box.ToString() );
            reader.SetRegion( box );

            // reader will try to load the uncompressed image region into buffer.
            // the call returns an error when buffer.Length is too small. For instance
            // one can call:
            // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
            // to get the exact size of minimum buffer
            if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
            {
                using (System.IO.Stream stream =

```

```

        System.IO.File.Open(@"tmp/frame.raw",
            System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

14.24 ExtractImageRegionWithLUT.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 * Furthermore we are applying the LUT on this image.
 * Special care should be taken in case the image is not PALETTE COLOR
 *
 * Usage:
 * $ bin/ExtractImageRegionWithLUT.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16loo.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
 * $ gdcminimg --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcviewer rgb.dcm
 */
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return 1;
        // Get file infos
        gdcm.File f = reader.GetFile();

        gdcm.LookupTable lut = reader.GetImage().GetLUT();

        // get some info about image
        UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
        PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
        int pixelsize = pf.GetPixelSize();

        // buffer to get the pixels

```

```

byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

// output buffer for the RGB decoded image:
byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (uint z = 0; z < dims[2]; z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
    //System.Console.WriteLine( box.ToString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
    {
        if( !lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ) )
        {
            throw new Exception("can't decode");
        }

        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/frame_rgb.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer2);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

14.25 ExtractOneFrame.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
* This small code shows how to use the gdcm.StreamImageReader API
* to read a single (whole) frame at a time
* The API allow extracting a smaller extent of the frame of course.
* It will write out the extracted frame in /tmp/frame.raw
*
* Usage:
* $ bin/ExtractOneFrame.exe input.dcm
*/
using System;
using gdcm;

public class ExtractOneFrame
{
    public static int Main(string[] args)
    {

```

```

string filename = args[0];

gdcm.StreamImageReader reader = new gdcm.StreamImageReader();

reader.SetFileName( filename );

if (!reader.ReadImageInformation()) return 1;
// Get file infos
gdcm.File f = reader.GetFile();

// get some info about image
UIntArrayType extent = ImageHelper.GetDimensionsValue(f);
//System.Console.WriteLine( extent[0] );
uint dimx = extent[0];
//System.Console.WriteLine( extent[1] );
uint dimy = extent[1];
//System.Console.WriteLine( extent[2] );
uint dimz = extent[2];
PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
int pixelsize = pf.GetPixelSize();
//System.Console.WriteLine( pixelsize );

// buffer to get the pixels
byte[] buffer = new byte[ dimx * dimy * pixelsize ];

for (int i = 0; i < dimz; i++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    reader.DefinePixelExtent(0, (ushort)dimx, 0, (ushort)dimy, (ushort)i, (ushort)(i+1));
    uint buf_len = reader.DefineProperBufferLength(); // take into account pixel size
    //System.Console.WriteLine( buf_len );
    if( buf_len > buffer.Length )
    {
        throw new Exception("buffer is too small for target");
    }

    if (reader.Read(buffer, (uint)buffer.Length))
    {
        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/frame.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}

```

14.26 FileAnonymize.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileAnonymize.exe input.dcm output.dcm

```

```

*/
using System;
using gdcms;

public class FileAnonymize
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        gdcms.FileAnonymizer fa = new gdcms.FileAnonymizer();
        fa.SetInputFileName( filename );
        fa.SetOutputFileName( outfilename );

        // Empty Operations
        // It will create elements, since those tags are non-registered public elements (2011):
        fa.Empty( new Tag(0x0008,0x1313) );
        fa.Empty( new Tag(0x0008,0x1317) );
        // Remove Operations
        // The following Tag are actually carefully chosen, since they refer to SQ:
        fa.Remove( new Tag(0x0008,0x2112) );
        fa.Remove( new Tag(0x0008,0x9215) );
        // Replace Operations
        // do not call replace operation on SQ attribute !
        fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
        fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

        if( !fa.Write() )
        {
            System.Console.WriteLine( "Could not write" );
            return 1;
        }

        return 0;
    }
}

```

14.27 FileChangeTS.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
* Simple C# example
*
* Shows multiple steps:
* Steps 1.
* Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
* image to be able to create the volume in memory Of course you can use any
* existing DICOM instead
*
* Step 2.
* Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
* At this point in time this makes the DICOM file invalid (truncated). But the
* next step will fix this.
*
* Step 3.
* Use C# to create a binary data which will represent our source object for
* image.
*
* Step 4.
* We use gdcms.FileStreamer to merge the template DICOM file from Step 2, with
* the binary data from Step 3. We decide to read a scanline at a time, but
* this can be read with any number of bytes. AppendToDataElement() will always
* do the proper computation.

```

```

*
* Step 5.
* We compress this gigantic file, into [JPEG Lossless, Non-Hierarchical,
* First-Order Prediction (Process 14 [Selection Value 1])]
*
* Usage:
* $ mono bin/FileChangeTS.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
*/
using System;
using System.IO;
using gdcmm;

public class FileChangeTS
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
        return encoding.GetBytes(str);
    }
    // Create a 256 x 256 Secondary Capture Image Storage
    static private void CreateSmallDICOM(string fileName)
    {
        using( var writer = new gdcmm.PixmapWriter() )
        {
            gdcmm.Pixmap img = writer.GetImage();
            img.SetNumberOfDimensions( 3 );
            img.SetDimension(0, 512 );
            img.SetDimension(1, 512 );
            img.SetDimension(2, 2 ); // fake a 3d volume
            PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2 );
            img.SetPhotometricInterpretation( pi );
            gdcmm.DataElement pixeldata = new gdcmm.DataElement( new gdcmm.Tag(0x7fe0,0x0010) );
            byte[] buffer = new byte[ 512 * 512 * 2 ];
            pixeldata.SetByteValue( buffer, new gdcmm.VL((uint)buffer.Length) );
            img.SetDataElement( pixeldata );

            gdcmm.File file = writer.GetFile();
            gdcmm.DataSet ds = file.GetDataSet();
            gdcmm.DataElement ms = new gdcmm.DataElement(new gdcmm.Tag(0x0008,0x0016));
            string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image Storage
            byte[] val = StrToByteArray(mediastorage);
            ms.SetByteValue( val, new gdcmm.VL( (uint)val.Length) );
            ds.Insert( ms );

            writer.SetFileName( fileName );
            writer.Write();
        }
    }
    static private void CreateBigDICOM(string fileName, string outfilename)
    {
        using( var ano = new gdcmm.FileAnonymizer() )
        {
            // The following is somewhat dangerous, do not try at home:
            string nframes = "1000";
            ano.Replace( new gdcmm.Tag(0x0028,0x0008), nframes );
            ano.SetInputFileName(fileName);
            ano.SetOutputFileName(outfilename);
            ano.Write(); // at this point the DICOM is invalid !
        }
    }
    static private void CreateDummyFile(string fileName, long length)
    {
        using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
        {
            // Looks like C# always init to 0 (fallocate ?)
            // For the purpose of the test we could add some random noise
            fileStream.SetLength(length);
        }
    }
    static private void ReadBytesIntoArray( byte[] array, FileStream source )
    {
        int numBytesToRead = array.Length;
        int numBytesRead = 0;
        while (numBytesToRead > 0)
        {
            // According to spec: Read() may return anything from 0 to numBytesToRead.
            int n = source.Read(array, numBytesRead, numBytesToRead);

            // Break when the end of the file is reached.
            if (n == 0)

```

```

        break;

        numBytesRead += n;
        numBytesToRead -= n;
    }
}

static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
{
    using ( var fs = new gdcM.FileStreamer() )
    {
        fs.SetTemplateFileName(dicomfn);
        fs.SetOutputFileName(outfn);
        gdcM.Tag pixeldata = new gdcM.Tag(0x7fe0, 0x0010);
        // FileStreamer support automatic checking of pixel data length
        // based on DICOM attributes, only if we say so:
        fs.CheckDataElement( pixeldata );
        // Declare we are working on Pixel Data attribute:
        fs.StartDataElement( pixeldata );
        using ( FileStream rawSource = new FileStream(rawdata,
            FileMode.Open, FileAccess.Read) )
        {
            byte[] bytes = new byte[512];
            // Only read one scanline at a time
            // We could have been reading more at once, if this is more efficient,
            // AppendToDataElement will do the logic in all cases.
            for( int i = 0; i < 512 * 1000; ++i )
            {
                // Read the source file into a byte array.
                ReadBytesIntoArray( bytes, rawSource );
                fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
            }
        }
        if( !fs.StopDataElement( pixeldata ) )
        {
            // Most likely an issue with Pixel Data Length computation:
            throw new Exception("StopDataElement failed");
        }
    }
}

static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
{
    using( var sfcts = FileChangeTransferSyntax.New() )
    {
        // Need to retrieve the actual C++ reference, to pass to
        // SimpleSubjectWatcher:
        FileChangeTransferSyntax fcts = sfcts.__ref__;
        SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
        gdcM.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 );
        fcts.SetTransferSyntax( ts );
        fcts.SetInputFileName( rawdicom );
        fcts.SetOutputFileName( jpegdicom );
        fcts.Change();
    }
}

public static int Main(string[] args)
{
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];

    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);

    return 0;
}
}

```

14.28 FileChangeTSLossy.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression]
 *
 * Usage:
 * $ bin/FileChangeTSLossy.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
 */
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
    public static byte[] StrToByteArray(string str)
    {
        {
            System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
            return encoding.GetBytes(str);
        }
    }
    // Create a 256 x 256 Secondary Capture Image Storage
    static private void CreateSmallDICOM(string fileName)
    {
        using( var writer = new gdcm.PixmapWriter() )
        {
            gdcm.Pixmap img = writer.GetImage();
            img.SetNumberOfDimensions( 3 );
            img.SetDimension(0, 512 );
            img.SetDimension(1, 512 );
            img.SetDimension(2, 2 ); // fake a 3d volume
            PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2 );
            img.SetPhotometricInterpretation( pi );
            gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
            byte[] buffer = new byte[ 512 * 512 * 2 ];
            pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
            img.SetDataElement( pixeldata );

            gdcm.File file = writer.GetFile();
            gdcm.DataSet ds = file.GetDataSet();
            gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
            string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image Storage
            byte[] val = StrToByteArray(mediastorage);
            ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
            ds.Insert( ms );

            writer.SetFileName( fileName );
            writer.Write();
        }
    }
}

```



```

static private void CreateBigDICOM(string fileName, string outfilename)
{
    using( var ano = new gdcm.FileAnonymizer() )
    {
        // The following is somewhat dangerous, do not try at home:
        string nframes = "1000";
        ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
        ano.SetInputFileName(fileName);
        ano.SetOutputFileName(outfilename);
        ano.Write(); // at this point the DICOM is invalid !
    }
}

static private void CreateDummyFile(string fileName, long length)
{
    using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
    {
        // Looks like C# always init to 0 (fallocate ?)
        // For the purpose of the test we could add some random noise
        fileStream.SetLength(length);
    }
}

static private void ReadBytesIntoArray( byte[] array, FileStream source )
{
    int numBytesToRead = array.Length;
    int numBytesRead = 0;
    while (numBytesToRead > 0)
    {
        // According to spec: Read() may return anything from 0 to numBytesToRead.
        int n = source.Read(array, numBytesRead, numBytesToRead);

        // Break when the end of the file is reached.
        if (n == 0)
            break;

        numBytesRead += n;
        numBytesToRead -= n;
    }
}

static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
{
    using ( var fs = new gdcm.FileStreamer() )
    {
        fs.SetTemplateFileName(dicomfn);
        fs.SetOutputFileName(outfn);
        gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
        // FileStreamer support automatic checking of pixel data length
        // based on DICOM attributes, only if we say so:
        fs.CheckDataElement( pixeldata );
        // Declare we are working on Pixel Data attribute:
        fs.StartDataElement( pixeldata );
        using (FileStream rawSource = new FileStream(rawdata,
            FileMode.Open, FileAccess.Read))
        {
            byte[] bytes = new byte[512];
            // Only read one scanline at a time
            // We could have been reading more at once, if this is more efficient,
            // AppendToDataElement will do the logic in all cases.
            for( int i = 0; i < 512 * 1000; ++i )
            {
                // Read the source file into a byte array.
                ReadBytesIntoArray( bytes, rawSource );
                fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
            }
        }
        if( !fs.StopDataElement( pixeldata ) )
        {
            // Most likely an issue with Pixel Data Length computation:
            throw new Exception("StopDataElement failed");
        }
    }
}

static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
{
    using( var sfcts = FileChangeTransferSyntax.New() )
    {
        // Need to retrieve the actual C++ reference, to pass to
        // SimpleSubjectWatcher:
        FileChangeTransferSyntax fcts = sfcts.__ref__();
        SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
        gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
        fcts.SetTransferSyntax( ts );
    }
}

```

```

        ImageCodec ic = fcts.GetCodec();
        JPEGCodec jpeg = JPEGCodec.Cast( ic );
        jpeg.SetLossless( false );
        jpeg.SetQuality( 50 ); // poor quality !

        fcts.SetInputFileName( rawdicom );
        fcts.SetOutputFileName( jpegdicom );
        fcts.Change();
    }
}
public static int Main(string[] args)
{
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];

    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);

    return 0;
}
}

```

14.29 FileStreaming.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileStreaming.exe gdcmData/CT_16b_signed-UsedBits13.dcm output.dcm
 *
 * The class will take care of group handling and will use the first available group:
 * (0009,0012) ?? (LO) [MYTEST] # 6,1 Private Creator
 */
using System;
using gdcm;

public class FileStreaming
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        gdcm.PrivateTag pt = new gdcm.PrivateTag( new gdcm.Tag(0x9,0x10), "MYTEST" );

        gdcm.FileStreamer fs = new gdcm.FileStreamer();
        fs.SetTemplateFileName( filename );
        fs.SetOutputFileName( outfilename );

        byte[] buffer = new byte[ 8192 ];
        uint len = (uint)buffer.Length;

        // In this example, we want that each newly created Private Attribute
        // contains at most 1000 bytes of incoming dataset.
        // We are also calling the function twice to check that appending mode is
        // working from one call to the other. The last element will have a length
        // of (2 * 8192) % 1000 = 384
    }
}

```

```

if( !fs.StartGroupDataElement( pt, 1000, 1 )
    || !fs.AppendToGroupDataElement( pt, buffer, len )
    || !fs.AppendToGroupDataElement( pt, buffer, len )
    || !fs.StopGroupDataElement( pt ) )
{
    System.Console.WriteLine( "Could not change private group" );
    return 1;
}

return 0;
}
}

```

14.30 GetArray.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class GetArray
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        Image image = reader.GetImage();

        PixelFormat pixeltype = image.GetPixelFormat();

        if( image.GetNumberOfDimensions() != 2 )
        {
            // For the purpose of the test, exit early on
            return 1;
        }
        uint dimx = image.GetDimension(0);
        uint dimy = image.GetDimension(1);
        uint npixels = dimx * dimy;
        //LookupTable lut = image.GetLUT();
        //uint rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        //byte[] rbuf = new byte[ rl ];
        //uint rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        //assert rl == rl2;

        //byte[] str1 = new byte[ image.GetBufferLength()];
        //image.GetBuffer( str1 );
        if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
            System.Console.WriteLine( "Processing UINT8 image type" );
            byte[] str1 = new byte[ npixels ];
            image.GetArray( str1 );
        }
        else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )

```

```

    {
        System.Console.WriteLine( "Processing INT16 image type" );
        short[] str1 = new short[ npixels ];
        image.GetArray( str1 );
    }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
    {
        System.Console.WriteLine( "Processing UINT16 image type" );
        ushort[] str1 = new ushort[ npixels ];
        image.GetArray( str1 );
    }
    else
    {
        //System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.toString() );
        System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.GetScalarTypeAsString() );
        // Get bytes
        byte[] str1 = new byte[ image.GetBufferLength()];
        image.GetBuffer( str1 );
    }

    return 0;
}
}

```

14.31 MpegVideoInfo.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This examples takes in a MPEG2 and write out a Video Endoscopic Imagae Storage
* encoded using MPEG2 @ Main Profile
* ref: http://chrisa.wordpress.com/2007/11/21/decoding-mpeg2-information/
* See also:
* http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
* http://cvs.linux.hr/cgi-bin/viewcvs.cgi/mpeg_mod/README.infompeg?view=markup
* http://www.guru-group.fi/~too/sw/m2vmp2cut/mpeg2info.c
*/

/*
* Provides information about an MPEG2 file, including the duration, frame rate, aspect
* ratio, and resolution. Good information about the MPEG2 file structure that helps
* explain parts of the code can be found here:
* http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
*
* Copyright (c) 2007 Chris Anderson (chrisa@wordpress.com)
*
* This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 2 of the License, or (at your option) any later version.
*
* This library is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*/
using System;
using System.IO;
using gdcm;

public class Mpeg2VideoInfo
{
    #region Member Variables
    private TimeSpan m_startTime = TimeSpan.Zero;
    private TimeSpan m_endTime = TimeSpan.Zero;
    private TimeSpan m_duration = TimeSpan.Zero;

```

```

private eAspectRatios m_aspectRatio = eAspectRatios.Invalid;
private eFrameRates m_frameRate = 0;
private int m_pictureWidth = 0;
private int m_pictureHeight = 0;
#endregion

#region Constants
private const byte PADDING_PACKET = 0xBE;
private const byte VIDEO_PACKET = 0xE0;
private const byte AUDIO_PACKET = 0xC0;
private const byte SYSTEM_PACKET = 0xBB;
private const byte TIMESTAMP_PACKET = 0xB8;
private const byte HEADER_PACKET = 0xB3;

private const int BUFFER_SIZE = 8162; // 8K buffer

private readonly static TimeSpan EMPTY_TIMESPAN = new TimeSpan(0, 0, -1);
#endregion

#region Enumerations
public enum eFrameRates
{
    Invalid,
    PulldownNTSC, // 24000d/1001d = 23.976 Hz
    Film, // 24 Hz
    PAL, // 25 Hz
    NTSC, // 30000d/1001d = 29.97 Hz
    DropFrameNTSC, // 30 Hz
    DoubleRatePAL, // 50 Hz
    DoubleRateNTSC, // 59.97 Hz
    DoubleRateDropFrameNTSC // 60 Hz
}

public enum eAspectRatios
{
    Invalid,
    VGA, // 1/1
    StandardTV, // 4/3
    LargeTV, // 16/9
    Cinema // 2.21/1
}
#endregion

#region Constructor
public Mpeg2VideoInfo(string file)
{
    ParseMpeg(file);
}
#endregion

#region Public Properties
public TimeSpan StartTime
{
    get { return m_startTime; }
}

public TimeSpan EndTime
{
    get { return m_endTime; }
}

public TimeSpan Duration
{
    get { return m_duration; }
}

public eAspectRatios AspectRatio
{
    get { return m_aspectRatio; }
}

public eFrameRates FrameRate
{
    get { return m_frameRate; }
}

public int PictureWidth
{
    get { return m_pictureWidth; }
}

```

```

    }

    public int PictureHeight
    {
        get { return m_pictureHeight; }
    }
    #endregion

    #region Private Functions
    private void ParseMpeg(string file)
    {
        FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
        BinaryReader br = new BinaryReader(fs);

        m_startTime = GetStartTimeStampInfo(br);
        m_endTime = GetEndTimeStampInfo(br);

        m_duration = m_endTime.Subtract(m_startTime);

        GetHeaderInfo(br);

        br.Close();
        fs.Close();
    }

    private TimeSpan GetStartTimeStampInfo(BinaryReader br)
    {
        TimeSpan startTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];

        br.BaseStream.Seek(0, SeekOrigin.Begin);

        while (startTime == EMPTY_TIMESPAN && br.BaseStream.Position < br.BaseStream.Length)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

            for (int offset = 0; offset < readBytes - 8; offset++)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    startTime = DecodeTimeStamp(timeStampEncoded);

                    if (startTime != EMPTY_TIMESPAN)
                        break;
                }
            }
        }

        return startTime;
    }

    private TimeSpan GetEndTimeStampInfo(BinaryReader br)
    {
        TimeSpan endTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];

        br.BaseStream.Seek(-BUFFER_SIZE, SeekOrigin.End);

        while (endTime == EMPTY_TIMESPAN && br.BaseStream.Position > BUFFER_SIZE)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

            for (int offset = readBytes - 8; offset >= 0; offset--)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    endTime = DecodeTimeStamp(timeStampEncoded);

                    if (endTime != EMPTY_TIMESPAN)
                        break;
                }
            }

            br.BaseStream.Seek(-BUFFER_SIZE * 2, SeekOrigin.Current);
        }

        return endTime;
    }

```

```

    }

    private TimeSpan DecodeTimeStamp(uint timeStampEncoded)
    {
        TimeSpan timeStamp = EMPTY_TIMESPAN;

        // Mask out the bits containing the property we are after, then
        // shift the data to the right to get its value
        int hour = (int)(timeStampEncoded & 0x7C000000) » 26; // Bits 31 -> 27
        int minute = (int)(timeStampEncoded & 0x03F00000) » 20; // Bits 26 -> 21
        int second = (int)(timeStampEncoded & 0x0007E000) » 13; // Bits 19 -> 14
        int frame = (int)(timeStampEncoded & 0x00001F80) » 7; // Bits 13 -> 8 - not used, but included for completeness

        timeStamp = new TimeSpan(hour, minute, second);
        return timeStamp;
    }

    private void GetHeaderInfo(BinaryReader br)
    {
        byte[] buffer = new byte[BUFFER_SIZE];

        br.BaseStream.Seek(0, SeekOrigin.Begin);
        br.Read(buffer, 0, BUFFER_SIZE);

        for (int offset = 0; offset < buffer.Length - 4; offset++)
        {
            if (IsStreamMarker(ref buffer, offset, HEADER_PACKET))
            {
                offset += 4; // Move to the data position which follows the stream header
                uint headerData = GetData(ref buffer, offset);

                // Mask out the bits containing the property we are after, then
                // shift the data to the right to get its value
                m_pictureWidth = (int)(headerData & 0xFFF00000) » 20;
                m_pictureHeight = (int)(headerData & 0x000FFF00) » 8;

                uint aspectRatioIndex = (headerData & 0x000000F0) » 4;
                uint fpsIndex = headerData & 0x0000000F;

                m_aspectRatio = (eAspectRatios)fpsIndex;
                m_frameRate = (eFrameRates)fpsIndex;

                break;
            }
        }
    }

    private uint GetData(ref byte[] buffer, int offset)
    {
        return (uint) ((buffer[offset] « 24) |
            (buffer[offset + 1] « 16) |
            (buffer[offset + 2] « 8) |
            (buffer[offset + 3]));
    }

    private bool IsStreamMarker(ref byte[] buffer, int offset, byte markerType)
    {
        return (buffer[offset] == 0x00 &&
            buffer[offset + 1] == 0x00 &&
            buffer[offset + 2] == 0x01 &&
            buffer[offset + 3] == markerType);
    }
    #endregion
    public static int Main(string[] args)
    {
        string file1 = args[0];
        Mpeg2VideoInfo info = new Mpeg2VideoInfo(file1);
        System.Console.WriteLine( info.StartTime );
        System.Console.WriteLine( info.EndTime );
        System.Console.WriteLine( info.Duration );
        System.Console.WriteLine( info.AspectRatio );
        System.Console.WriteLine( info.FrameRate );
        System.Console.WriteLine( info.PictureWidth );
        System.Console.WriteLine( info.PictureHeight );

        ImageReader r = new ImageReader();
        //Image image = new Image();
        Image image = r.GetImage();
        image.SetNumberOfDimensions( 3 );
    }

```

```

DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

System.IO.FileStream infile =
    new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
uint fsize = gdcm.PosixEmulation.FileSize(file1);

byte[] jstream = new byte[fsize];
infile.Read(jstream, 0 , jstream.Length);

SmartPtrFrag sq = SequenceOfFragments.New();
Fragment frag = new Fragment();
frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
sq.AddFragment( frag );
pixeldata.SetValue( sq.__ref__() );

// insert:
image.SetDataElement( pixeldata );

PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_PARTIAL_420 );
image.SetPhotometricInterpretation( pi );
// FIXME hardcoded:
PixelFormat pixeltype = new PixelFormat(3,8,8,7);
image.SetPixelFormat( pixeltype );

// FIXME hardcoded:
TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.MPEG2MainProfile);
image.SetTransferSyntax( ts );

image.SetDimension(0, (uint)info.PictureWidth);
image.SetDimension(1, (uint)info.PictureHeight);
image.SetDimension(2, 721);

ImageWriter writer = new ImageWriter();
gdcm.File file = writer.GetFile();
file.GetHeader().SetDataSetTransferSyntax( ts );
Anonymizer anon = new Anonymizer();
anon.SetFile( file );

MediaStorage ms = new MediaStorage( MediaStorage.MSType.VideoEndoscopicImageStorage);

UIDGenerator gen = new UIDGenerator();
anon.Replace( new Tag(0x0008,0x16), ms.GetString() );
anon.Replace( new Tag(0x0018,0x40), "25" );
anon.Replace( new Tag(0x0018,0x1063), "40.000000" );
anon.Replace( new Tag(0x0028,0x34), "4\\3" );
anon.Replace( new Tag(0x0028,0x2110), "01" );

writer.SetImage( image );
writer.SetFileName( "dummy.dcm" );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write" );
    return 1;
}

return 0;
}
}

```

14.32 NewSequence.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:

```



```

* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/NewSequence.exe gdcmData/012345.002.050.dcm out.dcm
*/
using System;
//using gdcm;

public class NewSequence
{
    public static byte[] StrToByteArray(string str)
    {
        System.Text.ASCIIEncoding encoding=new System.Text.ASCIIEncoding();
        return encoding.GetBytes(str);
    }

    public static int Main(string[] argv)
    {
        string file1 = argv[0];
        string file2 = argv[1];

        gdcm.Reader r = new gdcm.Reader();
        r.SetFileName( file1 );
        if ( ! r.Read() )
        {
            return 1;
        }

        gdcm.File f = r.GetFile();
        gdcm.DataSet ds = f.GetDataSet();
        // tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

        // Create a dataelement
        gdcm.DataElement de = new gdcm.DataElement(new gdcm.Tag(0x0010, 0x2180));
        string occ = "Occupation";
        de.SetByteValue( StrToByteArray(occ), new gdcm.VL((uint)occ.Length));
        de.SetVR(new gdcm.VR(gdcm.VR.VRType.SH));

        // Create an item
        gdcm.Item it = new gdcm.Item();
        it.SetVLToUndefined(); // Needed to not popup error message
        //it.InsertDataElement(de)
        gdcm.DataSet nds = it.GetNestedDataSet();
        nds.Insert(de);

        // Create a Sequence
        gdcm.SmartPtrSQ sq = gdcm.SequenceOfItems.New();
        sq.SetLengthToUndefined();
        sq.AddItem(it);

        // Insert sequence into data set
        gdcm.DataElement des = new gdcm.DataElement(new gdcm.Tag(0x0400,0x0550));
        des.SetVR(new gdcm.VR(gdcm.VR.VRType.SQ));
        des.SetValue(sq.__ref__());
        des.SetVLToUndefined();

        ds.Insert(des);

        gdcm.Writer w = new gdcm.Writer();
        w.SetFile( f );
        w.SetFileName( file2 );
        if ( !w.Write() )
            return 1;

        return 0;
    }
}

```

14.33 RescaleImage.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even

```

the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm rescaled.dcm
 */
using System;
using gdcm;

public class DecompressImage
{
    public static int Main(string[] args)
    {
        string file1 = args[0];
        ImageReader reader = new ImageReader();
        reader.SetFileName( file1 );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        Image image = reader.GetImage();
        PixelFormat pixeltype = image.GetPixelFormat();

        Rescaler r = new Rescaler();
        r.SetIntercept( 0 );
        r.SetSlope( 1.2 );
        r.SetPixelFormat( pixeltype );
        PixelFormat outputpt = new PixelFormat( r.ComputeInterceptSlopePixelFormat() );

        System.Console.WriteLine( "pixeltype" );
        System.Console.WriteLine( pixeltype.ToString() );
        System.Console.WriteLine( "outputpt" );
        System.Console.WriteLine( outputpt.ToString() );

        uint len = image.GetBufferLength();
        short[] input = new short[ len / 2 ]; // sizeof(short) == 2
        image.GetArray( input );

        double[] output = new double[ len / 2 ];
        r.Rescale( output, input, len );

        // First Pixel is:
        System.Console.WriteLine( "Input:" );
        System.Console.WriteLine( input[0] );

        System.Console.WriteLine( "Output:" );
        System.Console.WriteLine( output[0] );

        return 0;
    }
}

```

14.34 SendFileSCU.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

===== */
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm-gcc/bin

```

```

* $ mono bin/SendFileSCU.exe server port input.dcm
*/
using System;
using gdcm;

public class SendFileSCU
{
    public static int Main(string[] args)
    {
        string server = args[0];
        ushort port = ushort.Parse(args[1]);
        string filename = args[2];

        bool b = CompositeNetworkFunctions.CEcho( server, port );
        if( !b ) return 1;

        FilenamesType files = new FilenamesType();
        files.Add( filename );
        b = CompositeNetworkFunctions.CStore( server, port, files );
        if( !b ) return 1;

        return 0;
    }
}

```

14.35 SimplePrintPatientName.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
* $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
*/
/*
This example was provided by Jonathan Morra /jonmorra gmail com/
on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;

namespace GDCMTest
{
    class SimplePrintPatientName
    {
        static int Main(string[] args)
        {
            if (args.Length != 1)
            {
                Console.WriteLine("This program prints the patient name of a dicom file with gdcm");
                Console.WriteLine("Usage: [input.dcm]");
                return 1;
            }

            gdcm.Reader reader = new gdcm.Reader();
            reader.SetFileName(args[0]);
            bool ret = reader.Read();
            //TagSetType tst = new TagSetType();
            //tst.Add( new Tag(0x7fe0,0x10) );
            //bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
            if( !ret )
            {
                return 1;
            }
        }
    }
}

```

```

    }

    gdcml.File file = reader.GetFile();

    gdcml.StringFilter filter = new gdcml.StringFilter();
    filter.SetFile(file);
    string value = filter.ToString(new gdcml.Tag(0x0010, 0x0010));

    Console.WriteLine("Patient Name: " + value);
    return 0;
}
}

```

14.36 SortImage2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcml/debug-gcc/bin
 * $ mono bin/SortImage.exe gdcmlData/012345.002.050.dcm out.dcm
 */
using System;
using gdcml;

public class SortImage2
{
    bool mysort(DataSet ds1, DataSet ds2)
    {
        return false;
    }

    public static int Main(string[] args)
    {
        Sorter sorter = new Sorter();
        sorter.SetSortFunction( mysort );

        return 0;
    }
}

```

14.37 CStoreQtProgress.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small example show how one can use the virtual function
 * mechanism of the SimpleSubjectWatcher class to redirect progress

```

```

* report to a custom Qt classes
*
* http://doc.qt.nokia.com/latest/qprogressdialog.html
*
* Usage:
* CStoreQtProgress dicom.example.com 11112 gdcmlData/MR_Spectroscopy_SIEMENS_OF.dcm
*/

#include "gdcmlServiceClassUser.h"
#include "gdcmlSimpleSubjectWatcher.h"
#include "gdcmlProgressEvent.h"
#include "gdcmlDirectory.h"
#include "gdcmlPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcml {
/*
* This class is a little more complicated than what this example demonstrate
* This watcher is capable of handling nested progress. Since the Progress
* grows from [0 to 1] on a per file basis and we only have one instance of a
* watcher per association, we need some calculation to compute the global
* (total) progress
* In fact we simply divide the per-file progress by the number of files.
*
* This QtWatcher class will then update the progress bar according to the
* progress.
*/
class MyQtWatcher : public SimpleSubjectWatcher
{
    size_t nfiles;
    double progress;
    size_t index;
    double refprogress;
    QWidget* win;
    QProgressDialog* qtprogress;
public:
    MyQtWatcher(Subject* s, const char* comment = "", QWidget* w = NULL, QProgressDialog* p = NULL, size_t n = 1):
        SimpleSubjectWatcher(s, comment), nfiles(n), progress(0), index(0), refprogress(0), win(w), qtprogress(p) {}
    void ShowIteration()
    {
        index++;
        gdcml_assert( index <= nfiles );
        // update refprogress (we are moving to the next file)
        refprogress = progress;
    }
    void ShowProgress(Subject*, const Event& evt)
    {
        // Retrieve the ProgressEvent:
        const ProgressEvent& pe = dynamic_cast<const ProgressEvent*>(evt);
        // compute global progress:
        progress = refprogress + (1. / (double)nfiles) * pe.GetProgress();
        // Print Global and local progress to stdout:
        std::cout << "Global Progress: " << progress << " per file progress " << pe.GetProgress() << std::endl;
        //set progress value in the QtProgress bar
        int i = (int)(progress * 100 + 0.5); // round to next int
        qtprogress->setValue(i);
        win->show();
    }
    virtual void ShowDataSet(Subject* caller, const Event& evt)
    {
        (void)caller;
        (void)evt;
    }
};
} // end namespace gdcml

int main(int argc, char* argv[])
{
    if( argc < 4 )
    {
        std::cerr << argv[0] << " remote_server port filename" << std::endl;
        return 1;
    }
    QApplication a(argc, argv);

    std::ostringstream error_log;

```

```

gdcmm::Trace::SetErrorStream( error_log );

const char *remote = argv[1];
int portno = atoi(argv[2]);
const char *filename = argv[3];

QVBoxLayout* layout = new QVBoxLayout;
QWidget* win = new QWidget;

QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel", 0, 100);
progress->setWindowModality(Qt::WindowModal);

layout->addWidget(progress,Qt::AlignCenter);
win->setLayout(layout);

gdcmm::SmartPointer<gdcmm::ServiceClassUser> scup = new gdcmm::ServiceClassUser;
gdcmm::ServiceClassUser &scu = *scup;
//gdcmm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
// let's use a more complicated progress reported in this example
gdcmm::MyQtWatcher w( &scu, "QtWatcher", win, progress );

scu.SetHostname( remote );
scu.SetPort( (uint16_t)portno );
scu.SetTimeout( 1000 );
scu.SetCalledAETitle( "GDCM_STORE" );

if( !scu.InitializeConnection() )
{
    std::cerr << "Could not InitializeConnection" << std::endl;
    return 1;
}

gdcmm::Directory::FilenamesType filenames;
filenames.push_back( filename );

// setup the PC(s) based on the filenames:
gdcmm::PresentationContextGenerator generator;
if( !generator.GenerateFromFilenames(filenames) )
{
    std::cerr << "Could not GenerateFromFilenames" << std::endl;
    return 1;
}

// Setup PresentationContext(s)
scu.SetPresentationContexts( generator.GetPresentationContexts() );

// Start ASSOCIATION
if( !scu.StartAssociation() )
{
    std::cerr << "Could not Start" << std::endl;
    return 1;
}

// Send C-STORE
if( !scu.SendStore( filename ) )
{
    std::cerr << "Could not Store" << std::endl;
    std::cerr << "Error log is:" << std::endl;
    std::cerr << error_log.str() << std::endl;
    return 1;
}

// Stop ASSOCIATION
if( !scu.StopAssociation() )
{
    std::cerr << "Could not Stop" << std::endl;
    return 1;
}

win->show();

return a.exec();
}

```

14.38 ChangePrivateTags.cxx

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```
=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmPrivateTag.h"

int main(int argc, char* argv[] )
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " path/to/05148044-mr-siemens-avanto-syngo.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if ( ! reader.Read() )
    {
        return 1;
    }

    // (0029,0010) LO [SIEMENS CSA HEADER] # 18,1 Private Creator
    // (0029,0011) LO [SIEMENS MEDCOM HEADER ] # 22,1 Private Creator
    // (0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22,1 Private Creator
    // [...]
    // (0029,1018) CS [MR] # 2,1 CSA Series Header Type
    // (0029,1134) CS [DB TO DICOM ] # 12,1 PMTF Information 4
    // (0029,1260) LO [com ] # 4,1 Series Workflow Status

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Declare private tag we need to find:
    gdcm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
    gdcm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );
    gdcm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );

    const char str1[] = "GDCM was here 3!";
    if ( !ds.FindDataElement( pt1 ) ) return 1;
    gdcm::DataElement de1 = ds.GetDataElement( pt1 ); // Convert Private tag, into actual DataElement
    std::cout << de1 << std::endl;
    de1.SetByteValue( str1, (uint32_t)strlen(str1) );
    ds.Replace( de1 );

    const char str2[] = "GDCM was here 2!";
    if ( !ds.FindDataElement( pt2 ) ) return 1;
    gdcm::DataElement de2 = ds.GetDataElement( pt2 );
    std::cout << de2 << std::endl;
    de2.SetByteValue( str2, (uint32_t)strlen(str2) );
    ds.Replace( de2 );

    const char str3[] = "GDCM was here 3!";
    if ( !ds.FindDataElement( pt3 ) ) return 1;
    gdcm::DataElement de3 = ds.GetDataElement( pt3 );
    std::cout << de3 << std::endl;
    de3.SetByteValue( str3, (uint32_t)strlen(str3) );
    ds.Replace( de3 );

    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfile );
    if ( !writer.Write() )
    {
        return 1;
    }

    return 0;
}
```

14.39 ChangeSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmSmartPointer.h"
#include "gdcmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is the exact C++ translation of the original python example: ManipulateSequence.py
*/

int main(int argc, char* argv[] )
{
    if( argc < 3 )
    {
        std::cerr << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    gdcm::Tag tsis(0x0008,0x2112); // SourceImageSequence
    if ( ds.FindDataElement( tsis ) )
    {
        const gdcm::DataElement &sis = ds.GetDataElement( tsis );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqsis = sis.GetValueAsSQ();
        if ( sqsis && sqsis->GetNumberOfItems() )
        {
            gdcm::Item &item1 = sqsis->GetItem(1);
            gdcm::DataSet &nstedds = item1.GetNestedDataSet();
            gdcm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
            if( nstedds.FindDataElement( tprcs ) )
            {
                const gdcm::DataElement &prcs = nstedds.GetDataElement( tprcs );
                gdcm::SmartPointer<gdcm::SequenceOfItems> sqprcs = prcs.GetValueAsSQ();
                if ( sqprcs && sqprcs->GetNumberOfItems() )
                {
                    gdcm::Item &item2 = sqprcs->GetItem(1);
                    gdcm::DataSet &nstedds2 = item2.GetNestedDataSet();
                    // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
                    gdcm::Tag tcm(0x0008,0x0104);
                    if( nstedds2.FindDataElement( tcm ) )
                    {
                        gdcm::DataElement cm = nstedds2.GetDataElement( tcm );
                        std::string mystr = "GDCM was here";
                        cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                        nstedds2.Replace( cm );
                    }
                }
            }
        }
    }

    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfile );

```



```

if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

14.40 CheckBigEndianBug.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmmconv, you would like to know if the conversion process is acceptable
 * sometime a vbndiff is acceptable, sometime it is not. In the case of the famous Philips
 * Little/Big Endian Explicit Transfer Syntax it is not easy to compare two files. However
 * this only impact byte ordering, thus we can compute byte-independant information to still
 * compare the files.
 */

#include "gdcmmImageReader.h"
#include "gdcmmImage.h"
#include "gdcmmWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmSystem.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input1.dcm input2.dcm" << std::endl;
        return 1;
    }
    const char *filename1 = argv[1];
    const char *filename2 = argv[2];

    gdcmm::ImageReader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        std::cerr << "Could not read: " << filename1 << std::endl;
        return 1;
    }

    gdcmm::ImageReader reader2;
    reader2.SetFileName( filename2 );
    if( !reader2.Read() )
    {
        std::cerr << "Could not read: " << filename2 << std::endl;
        return 1;
    }

    // TODO: need a DataSet== operator implementation

    std::cout << "Both files can be read and looks like DICOM" << std::endl;

    size_t s1 = gdcmm::System::FileSize(filename1);
    size_t s2 = gdcmm::System::FileSize(filename2);

    if( s1 != s2 )
    {

```

```

    std::cout << "Size mismatch: " << s1 << " != " << s2 << std::endl;
    return 1;
}
else
{
    std::cout << "Size match: " << s1 << " = " << s2 << std::endl;
}

std::ifstream is1( filename1, std::ios::binary );
char *buffer1 = new char[s1];
is1.read(buffer1, s1);

std::ifstream is2( filename2, std::ios::binary );
char *buffer2 = new char[s2];
is2.read(buffer2, s2);

gdcmm_assert( s1 == s2 );
if( memcmp(buffer1, buffer2, s1 ) == 0 )
{
    std::cout << "memcmp succeed ! File are bit identical" << std::endl;
}
else
{
    std::cout << "memcmp failed!" << std::endl;
}

// Hum...memcmp failed, for big endian/ little endian inversion the histogram of bytes
// should still be the same. So let's compute it
// buffer2[0] = 1; // let's make the test fail
std::multiset<char> set1( buffer1, buffer1 + s1 );
std::multiset<char> set2( buffer2, buffer2 + s2 );

if( set1 == set2 )
{
    std::cout << "set1 == set2. Byte histogram seems valid" << std::endl;
}
else
{
    std::cout << "set1 != set2" << std::endl;
}
delete[] buffer1;
delete[] buffer2;

return 0;
}

```

14.41 ClinicalTrialAnnotate.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Dummy implementation of C.7.1.3 Clinical Trial Subject Module
* Usage:
* ClinicalTrialAnnotate gdcmmData/012345.002.050.dcm out.dcm
*/

#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmAnonymizer.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )

```

```

{
    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
}
const char *filename = argv[1];
const char *outfilename = argv[2];

gdcm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}

// The output of gdcm::Reader is a gdcm::File
//gdcm::File &file = reader.GetFile();

// the dataset is the the set of element we are interested in:
//gdcm::DataSet &ds = file.GetDataSet();

gdcm::Anonymizer ano;
ano.SetFile( reader.GetFile() );
ano.RemoveGroupLength();
ano.RemovePrivateTags();

// PS 3.3 - 2008
// C.7.1.3 Clinical Trial Subject Module
// <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial Sponsor Name"/>
ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
// <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial Protocol ID"/>
ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial Protocol ID" );
// <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial Protocol Name"/>
ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial Protocol Name" );
// <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial Site ID"/>
ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID" );
// <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial Site Name"/>
ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
// <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
// <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>
ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );

gdcm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfile );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

14.42 CompressImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

/*=====*/
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"

```

```

#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include "gdcmImageChangeTransferSyntax.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    //gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    //gdcm::DataSet &ds = file.GetDataSet();

    gdcm::Image &image = reader.GetImage();
    // image.SetSpacing(0, 0.1);
    // image.SetSpacing(1, 0.2);
    image.Print( std::cout );

    gdcm::ImageChangeTransferSyntax change;
    change.SetTransferSyntax( gdcm::TransferSyntax::JPEG2000Lossless );
    change.SetTransferSyntax( gdcm::TransferSyntax::JPEGLosslessProcess14_1 );
    //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
    //change.SetTransferSyntax( image.GetTransferSyntax() );
    change.SetInput( image );
    bool b = change.Change();
    if( !b )
    {
        std::cerr << "Could not change the Transfer Syntax" << std::endl;
        return 1;
    }

    //std::ofstream out( outfile, std::ios::binary );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

14.43 ConvertToQImage.cxx

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

===== */
/*
 * This example shows how to setup the pipeline from a gdcm::ImageReader into a
 * Qt QImage data structure.
 * It only handles 2D image.
 *
 * Ref:
 * http://doc.trolltech.com/4.5/qimage.html
 *
 * Usage:
 * ConvertToQImage gdcmData/012345.002.050.dcm output.png
 *
 * Thanks:
 * Sylvain ADAM (sylvain51 hotmail com) for contributing this example
 */

#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>

bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer, QImage* &imageQt)
{
    unsigned int* dimension = gimage.GetDimensions();

    unsigned int dimX = dimension[0];
    unsigned int dimY = dimension[1];

    gimage.GetBuffer(buffer);

    // Let's start with the easy case:
    if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::RGB )
    {
        if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8 )
        {
            return false;
        }
        unsigned char *ubuffer = (unsigned char*)buffer;
        // QImage::Format_RGB888 13 The image is stored using a 24-bit RGB format (8-8-8).
        imageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX, QImage::Format_RGB888);
    }
    else if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::MONOCHROME2 )
    {
        if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8 )
        {
            // We need to copy each individual 8bits into R / G and B:
            unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
            unsigned char *pubuffer = ubuffer;
            for(unsigned int i = 0; i < dimX*dimY; i++)
            {
                *pubuffer++ = *buffer;
                *pubuffer++ = *buffer;
                *pubuffer++ = *buffer++;
            }

            imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
        }
        else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16 )
        {
            // We need to copy each individual 16bits into R / G and B (truncate value)
            short *buffer16 = (short*)buffer;
            unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
            unsigned char *pubuffer = ubuffer;
            for(unsigned int i = 0; i < dimX*dimY; i++)
            {
                // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could simply do:
                // *pubuffer++ = *buffer16;
                // *pubuffer++ = *buffer16;
                // *pubuffer++ = *buffer16;
                // instead do it right:
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                buffer16++;
            }

            imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
        }
        else
        {
            std::cerr << "Pixel Format is: " << gimage.GetPixelFormat() << std::endl;
        }
    }
}

```

```

        return false;
    }
}
else
{
    std::cerr << "Unhandled PhotometricInterpretation: " << gimage.GetPhotometricInterpretation() << std::endl;
    return false;
}

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader ir;
    ir.SetFileName( filename );
    if(!ir.Read())
    {
        //Read failed
        return 1;
    }

    std::cout<<"Getting image from ImageReader..."<<std::endl;

    const gdcm::Image &gimage = ir.GetImage();
    std::vector<char> vbuffer;
    vbuffer.resize( gimage.GetBufferLength() );
    char *buffer = &vbuffer[0];

    QImage *imageQt = NULL;
    if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
    {
        return 1;
    }

    QImageWriter writer;
    writer.setFormat("png");
    writer.setFileName( outfile );
    if( !writer.write( *imageQt ) )
    {
        return 1;
    }

    return 0;
}

```

14.44 CreateARGBImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* http://www.w3.org/Graphics/PNG/inline-alpha.html
* alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
* $ convert alphatest.png alphatest.rgba
*/

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"

```

```

#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.rgb output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename, std::ios::binary);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;
    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::ARGB;
    image.SetPhotometricInterpretation( pi );
    image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
    pixeldata.SetByteValue( buf, (uint32_t)len );
    image.SetDataElement( pixeldata );

    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;

    return 0;
}

```

14.45 CreateCMYKImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* http://www.w3.org/Graphics/PNG/inline-alpha.html
* alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
*
* $ convert alphatest.png alphatest.cmyk
*/

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"

```

```

#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.cmyk output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename, std::ios::binary);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;
    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::CMYK;
    image.SetPhotometricInterpretation( pi );
    image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
    pixeldata.SetByteValue( buf, (uint32_t)len );
    image.SetDataElement( pixeldata );

    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;

    return 0;
}

```

14.46 CreateJPIPDataSet.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example was created during the GSOC 2011 project for
 * JPIP
 */
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])

```



```

{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];

    gdcmm::Writer w;
    gdcmm::File &file = w.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfile );

    file.GetHeader().SetDataSetTransferSyntax( gdcmm::TransferSyntax::JPIPReferenced );

    gdcmm::Anonymizer anon;
    anon.SetFile( file );

    gdcmm::MediaStorage ms = gdcmm::MediaStorage::SecondaryCaptureImageStorage;

    gdcmm::UIDGenerator gen;
    anon.Replace( gdcmm::Tag(0x0008,0x16), ms.GetString() );
    std::cout << ms.GetString() << std::endl;
    anon.Replace( gdcmm::Tag(0x0008,0x18), gen.Generate() );
    //
    anon.Replace( gdcmm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
    anon.Replace( gdcmm::Tag(0x0010,0x20), "012345" );
    anon.Empty( gdcmm::Tag(0x0010,0x30) );
    anon.Empty( gdcmm::Tag(0x0010,0x40) );
    anon.Empty( gdcmm::Tag(0x0008,0x20) );
    anon.Empty( gdcmm::Tag(0x0008,0x30) );
    anon.Empty( gdcmm::Tag(0x0008,0x90) );
    anon.Empty( gdcmm::Tag(0x0020,0x10) );
    anon.Empty( gdcmm::Tag(0x0020,0x11) );
    anon.Empty( gdcmm::Tag(0x0008,0x50) );
    anon.Empty( gdcmm::Tag(0x0020,0x0013) );
    anon.Replace( gdcmm::Tag(0x0020,0xd), gen.Generate() );
    anon.Replace( gdcmm::Tag(0x0020,0xe), gen.Generate() );
    anon.Replace( gdcmm::Tag(0x0008,0x64), "WSD " );
    anon.Replace( gdcmm::Tag(0x0008,0x60), "OT" );

    gdcmm::Attribute<0x0028,0x7FE0> at;
    at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2" );
    ds.Insert( at.GetAsDataElement() );

    // Need to retrieve the PixelFormat information from the given file

    if ( !w.Write() )
    {
        std::cerr << "Could not write: " << outfile << std::endl;
        return 1;
    }
    return 0;
}

```

14.47 DeriveSeries.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmFileDerivation.h"

```

```

#include "gdcmUIDGenerator.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char * ref = argv[1];
    const char * in  = argv[2];

    gdcm::Reader r1;
    r1.SetFileName( ref );
    if( !r1.Read() ) return 1;

    gdcm::Reader r2;
    r2.SetFileName( in );
    if( !r2.Read() ) return 1;

    // Fix Spatial info:
    gdcm::DataSet & ds1 = r1.GetFile().GetDataSet();
    gdcm::File & file2 = r2.GetFile();
    gdcm::DataSet & ds2 = file2.GetDataSet();
    //gdcm::Attribute<0x8,0x8> img_type = { "ORIGINAL", "PRIMARY" };
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0008) ));
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0032) ));
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0037) ));
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0088) )); // Spacing between slices
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0013) )); // Instance Number
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x5100) )); // Patient Position
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0050) )); // Slice Thickness
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0070) )); // Manufacturer
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0081) )); // Echo Time
    ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x1041) )); // Slice Location

    gdcm::Attribute<0x8,0x16> sopclassuid;
    sopclassuid.SetFromDataSet( ds1 );
    gdcm::Attribute<0x8,0x18> sopinstanceuid;
    sopinstanceuid.SetFromDataSet( ds1 );

    // Step 2: DERIVED object
    gdcm::FileDerivation fd;
    fd.AddReference( sopclassuid.GetValue(), sopinstanceuid.GetValue() );

    // http://dicom.nema.org/MEDICAL/dicom/current/output/chtml/part16/chapter_D.html#DCM_121321
    // CID 7202 "Source Image Purposes of Reference"
    // DCM 121321 "Mask image for image processing operation"
    fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121321 );
    // CID 7203 "Image Derivation"
    // DCM 113047 "Pixel by pixel mask"
    fd.SetDerivationCodeSequenceCodeValue( 113047 );
    fd.SetFile( file2 );
    // If all Code Value are ok the filter will execute properly
    if( !fd.Derive() )
    {
        std::cerr << "Sorry could not derive using input info" << std::endl;
        return 1;
    }

    gdcm::Writer w;
    w.SetFile( r2.GetFile() );
    w.SetFileName( "derived.dcm" );
    if( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

14.48 DiffFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre
 All rights reserved.
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
  {
    std::cerr « argv[0] « " input1.dcm input2.dcm" « std::endl;
    return 1;
  }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];

  gdcm::Reader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
  {
    return 1;
  }

  gdcm::Reader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
  {
    return 1;
  }

  const gdcm::File &file1 = reader1.GetFile();
  const gdcm::File &file2 = reader2.GetFile();

  const gdcm::DataSet &ds1 = file1.GetDataSet();
  const gdcm::DataSet &ds2 = file2.GetDataSet();

  gdcm::DataSet::ConstIterator it1 = ds1.Begin();
  gdcm::DataSet::ConstIterator it2 = ds2.Begin();

  const gdcm::DataElement &de1 = *it1;
  const gdcm::DataElement &de2 = *it2;
  if( de1 == de2 )
  {
  }
  while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
  {
    ++it1;
    ++it2;
  }

  if( it1 != ds1.End() || it2 != ds2.End() )
  {
    std::cerr « "Problem with:" « std::endl;
    if( it1 != ds1.End() )
    {
      std::cerr « "ds1: " « *it1 « std::endl;
    }
    if( it2 != ds2.End() )
    {
      std::cerr « "ds2: " « *it2 « std::endl;
    }
    return 1;
  }
  return 0;
}

```

14.49 DiscriminateVolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```
=====*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>

/*
 * The following example is a basic sorted which should work in generic cases.
 * It sort files based on:
 * Study Instance UID
 * Series Instance UID
 * Frame of Reference UID
 * Image Orientation (Patient)
 * Image Position (Patient) (Sorting based on IPP + IOP)
 */

namespace gdcm {
    const Tag t1(0x0020,0x000d); // Study Instance UID
    const Tag t2(0x0020,0x000e); // Series Instance UID
    const Tag t3(0x0020,0x0052); // Frame of Reference UID
    const Tag t4(0x0020,0x0037); // Image Orientation (Patient)

    class DiscriminateVolume
    {
    private:
        std::vector< Directory::FileNamesType > SortedFiles;
        std::vector< Directory::FileNamesType > UnsortedFiles;

        Directory::FileNamesType GetAllFileNamesFromTagToValue(
            Scanner const & s, Directory::FileNamesType const & filesubset, Tag const & t, const char *valueref)
        {
            Directory::FileNamesType theReturn;
            if( valueref )
            {
                size_t len = strlen( valueref );
                Directory::FileNamesType::const_iterator file = filesubset.begin();
                for( ; file != filesubset.end(); ++file )
                {
                    const char *filename = file->c_str();
                    const char * value = s.GetValue(filename, t);
                    if( value && strcmp(value, valueref, len) == 0 )
                    {
                        theReturn.push_back( filename );
                    }
                }
            }
            return theReturn;
        }
    }

    void ProcessAIOP(Scanner const & , Directory::FileNamesType const & subset, const char *iopval)
    {
        std::cout << "IOP: " << iopval << std::endl;
        IPPSorter ipp;
        ipp.SetComputeZSpacing( true );
        ipp.SetZSpacingTolerance( 1e-3 ); // ??
        bool b = ipp.Sort( subset );
        if( !b )
        {
            // If you reach here this means you need one more parameter to discriminiat this
            // series. Eg. T1 / T2 intertwined. Multiple Echo (0018,0081)
            std::cerr << "Failed to sort: " << subset.begin()->c_str() << std::endl;
            for(
                Directory::FileNamesType::const_iterator file = subset.begin();
                file != subset.end(); ++file )
            {
                std::cerr << *file << std::endl;
            }
            UnsortedFiles.push_back( subset );
            return ;
        }
        ipp.Print( std::cout );
    }
}
```

```

    SortedFiles.push_back( ipp.GetFileNames() );
}

void ProcessAFrameOfRef(Scanner const & s, Directory::FileNamesType const & subset, const char * frameuid)
{
    // In this subset of files (belonging to same series), let's find those
    // belonging to the same Frame ref UID:
    Directory::FileNamesType files = GetAllFileNamesFromTagToValue(
        s, subset, t3, frameuid);

    std::set< std::string > iopset;

    for(
        Directory::FileNamesType::const_iterator file = files.begin();
        file != files.end(); ++file)
    {
        //std::cout << *file << std::endl;
        const char * value = s.GetValue(file->c_str(), gdcmm::t4 );
        gdcmm_assert( value );
        iopset.insert( value );
    }
    size_t n = iopset.size();
    if ( n == 0 )
    {
        gdcmm_assert( files.empty() );
        return;
    }

    std::cout << "Frame of Ref: " << frameuid << std::endl;
    if ( n == 1 )
    {
        ProcessAIOP(s, files, iopset.begin()->c_str() );
    }
    else
    {
        const char *f = files.begin()->c_str();
        std::cerr << "More than one IOP: " << f << std::endl;
        // Make sure that there is actually 'n' different IOP
        gdcmm::DirectionCosines ref;
        gdcmm::DirectionCosines dc;
        for(
            std::set< std::string >::const_iterator it = iopset.begin();
            it != iopset.end(); ++it )
        {
            ref.SetFromString( it->c_str() );
            for(
                Directory::FileNamesType::const_iterator file = files.begin();
                file != files.end(); ++file)
            {
                std::string value = s.GetValue(file->c_str(), gdcmm::t4 );
                if( value != it->c_str() )
                {
                    dc.SetFromString( value.c_str() );
                    const double crossdot = ref.CrossDot(dc);
                    const double eps = std::fabs( 1. - crossdot );
                    if( eps < 1e-6 )
                    {
                        std::cerr << "Problem with IOP discrimination: " << file->c_str()
                            << " " << it->c_str() << std::endl;
                        return;
                    }
                }
            }
        }
        // If we reach here this means there is actually 'n' different IOP
        for(
            std::set< std::string >::const_iterator it = iopset.begin();
            it != iopset.end(); ++it )
        {
            const char *iopvalue = it->c_str();
            Directory::FileNamesType iopfiles = GetAllFileNamesFromTagToValue(
                s, files, t4, iopvalue );
            ProcessAIOP(s, iopfiles, iopvalue );
        }
    }
}

void ProcessASeries(Scanner const & s, const char * seriesuid)
{
    std::cout << "Series: " << seriesuid << std::endl;
    // let's find all files belonging to this series:

```

```

Directory::FileNamesType seriesfiles = GetAllFileNamesFromTagToValue(
    s, s.GetFileNames(), t2, seriesuid);

gdcmm::Scanner::ValuesType vt3 = s.GetValues(t3);
for(
    gdcmm::Scanner::ValuesType::const_iterator it = vt3.begin()
    ; it != vt3.end(); ++it )
{
    ProcessAFrameOfRef(s, seriesfiles, it->c_str());
}

void ProcessAStudy(Scanner const & s, const char * studyuid)
{
    std::cout << "Study: " << studyuid << std::endl;
    gdcmm::Scanner::ValuesType vt2 = s.GetValues(t2);
    for(
        gdcmm::Scanner::ValuesType::const_iterator it = vt2.begin()
        ; it != vt2.end(); ++it )
    {
        ProcessASeries(s, it->c_str());
    }
}

public:

void Print( std::ostream & os )
{
    os << "Sorted Files: " << std::endl;
    for(
        std::vector< Directory::FileNamesType >::const_iterator it = SortedFiles.begin();
        it != SortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FileNamesType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
    os << "Unsorted Files: " << std::endl;
    for(
        std::vector< Directory::FileNamesType >::const_iterator it = UnsortedFiles.begin();
        it != UnsortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FileNamesType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
}

std::vector< Directory::FileNamesType > const & GetSortedFiles() const { return SortedFiles; }
std::vector< Directory::FileNamesType > const & GetUnsortedFiles() const { return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
    gdcmm::Scanner::ValuesType vt1 = s.GetValues( gdcmm::t1 );
    for(
        gdcmm::Scanner::ValuesType::const_iterator it = vt1.begin()
        ; it != vt1.end(); ++it )
    {
        ProcessAStudy( s, it->c_str() );
    }
}

};

} // namespace gdcmm

int main(int argc, char *argv[])
{
    std::string dir1;
    if( argc < 2 )
    {
        const char *extradataroot = nullptr;

```

```

#ifdef GDCM_BUILD_TESTING
    extradataroot = gdcml::Testing::GetDataExtraRoot();
#endif
if( !extradataroot )
{
    return 1;
}
dir1 = extradataroot;
dir1 += "/gdcmlSampleData/ForSeriesTesting/VariousIncidences/ST1";
}
else
{
    dir1 = argv[1];
}

gdcml::Directory d;
d.Load( dir1, true ); // recursive !

gdcml::Scanner s;
s.AddTag( gdcml::t1 );
s.AddTag( gdcml::t2 );
s.AddTag( gdcml::t3 );
s.AddTag( gdcml::t4 );
bool b = s.Scan( d.GetFilesNames() );
if( !b )
{
    std::cerr << "Scanner failed" << std::endl;
    return 1;
}

gdcml::DiscriminateVolume dv;
dv.ProcessIntoVolume( s );
dv.Print( std::cout );

return 0;
}

```

14.50 DumpADAC.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* the goal of this example is to mimic the behavior of disp_img_header
* see http://www.gmccorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
*/
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
    uint16_t key;
    const char *name;
};

dict Array[] = {
    { 0x01, "Patient name" },

```

```

{ 0x02, "Patient ID" },
{ 0x03, "Patient sex" },
{ 0x04, "Patient age" },
{ 0x05, "Patient height" },
{ 0x06, "Patient weight" },
{ 0x07, "Exam date" },
{ 0x08, "Dose admin. time" },
{ 0x09, "Unique exam key" },
{ 0x0a, "Exam procedure" },
{ 0x0b, "Referring physician" },
{ 0x0c, "Attending physician" },
{ 0x0d, "Imaging modality" },
{ 0x0e, "Hospital ID" },
{ 0x0f, "Histogram crv file" },
{ 0x10, "Acq. start time" },
{ 0x11, "Object data type" },
{ 0x12, "Image viewid" },
{ 0x13, "Imaging device name" },
{ 0x14, "Device serial number" },
{ 0x15, "Collimator" },
{ 0x16, "Software version" },
{ 0x17, "Radiopharmaceutical #1" },
{ 0x18, "Energy window #1 center" },
{ 0x19, "Radiopharmaceutical #2" },
{ 0x1a, "Energy window #1 width" },
{ 0x1b, "Isotope imaging mode" },
{ 0x1c, "Energy window #2 center" },
{ 0x1d, "Energy window #2 width" },
{ 0x1e, "Energy window #3 center" },
{ 0x1f, "Energy window #3 width" },
{ 0x20, "Energy window #4 center" },
{ 0x21, "Energy window #4 width" },
{ 0x22, "??Energy window #5 center" },
{ 0x23, "??Energy window #5 width" },
{ 0x24, "Patient orientation" },
{ 0x25, "Spatial resolution" },
{ 0x26, "Slice thickness" },
{ 0x27, "Image X dimension" },
{ 0x28, "Image Y dimension" },
{ 0x29, "Image Z dimension" },
{ 0x2a, "Image pixel width" },
{ 0x2b, "Uniformity corr. file" },
{ 0x2c, "Acquisition zoom factor" },
{ 0x2d, "Total counts in set" },
{ 0x2e, "Time / frame" },
{ 0x2f, "Total acq. time" },
{ 0x30, "Maximum pixel value" },
{ 0x31, "Minimum pixel value" },
{ 0x32, "R-R interval time" },
{ 0x33, "Percent of cycle imaged" },
{ 0x34, "# of cycles accepted" },
{ 0x35, "# of cycles rejected" },
{ 0x36, "Approximate ED frame" },
{ 0x37, "Approximate ES frame" },
{ 0x38, "Approximate EF" },
{ 0x39, "Starting angle" },
{ 0x3a, "Degrees of rotation" },
{ 0x3b, "Direction of rotation" },
{ 0x3c, "Cont. or step/shoot" },
{ 0x3d, "Lim recon start frame" },
{ 0x3e, "Upper window grey shade" },
{ 0x3f, "Lower lvl grey shade" },
{ 0x40, "Associated color map" },
{ 0x41, "Custom color map file" },
{ 0x42, "Manipulated image" },
{ 0x43, "Axis of rotation corr." },
{ 0x44, "Reorientation azimuth" },
{ 0x45, "Reorientation elevation" },
{ 0x46, "Filter type" },
{ 0x47, "Filter order" },
{ 0x48, "Filter cutoff frequency" },
{ 0x49, "Reconstruction type" },
{ 0x4a, "Attenuation coefficient" },
{ 0x4b, "Associated parent file" },
{ 0x4c, "Unique patient key" },
{ 0x52, "Normalization crv file" },
{ 0x53, "Unique object key" },
{ 0x54, "This phase of VFR is" },
{ 0x55, "True color value" },
{ 0x56, "# of sets of x,y,z grps" },
{ 0x57, "Scale factor of set" },

```



```

{ 0x6d, "Date of birth" },
{ 0x6e, "Directional orientation" },
{ 0x6f, "Number of VFR studies" },
{ 0x70, "R-R low tolerance" },
{ 0x71, "R-R high tolerance" },
{ 0x72, "Prog specific results:" },

{ 0x99, nullptr }
};

void printname( int , int , uint16_t v )
{
    if( v == 0x1 )
    {
        std::cout << "DATABASE PARAMETERS" << std::endl;
        std::cout << "_____ " << std::endl;
    }
    else if( v == 0x27 )
    {
        std::cout << "IMAGE PARAMETERS" << std::endl;
        std::cout << "_____ " << std::endl;
    }
    else if( v == 0x13 )
    {
        std::cout << "EXTRA PARAMETERS" << std::endl;
        std::cout << "_____ " << std::endl;
    }
    else if( v == 0x2e )
    {
        std::cout << "*** NOT CURRENTLY USED : " << std::endl;
    }
    static const unsigned int n = sizeof( Array ) / sizeof( *Array ) - 1;
    for( unsigned int i = 0; i < n; ++i )
    {
        if( v == Array[i].key )
        {
            std::cout << /*" " << std::dec << len << ", " << mult << " " << */ Array[i].name;
            std::cout << " : ";
            return;
        }
    }
    std::cout << /*"\t#" << std::dec << len << ", " << mult << */ std::hex << v << "\t: ";
}

uint16_t readint16(std::istream &is )
{
    uint16_t val;
    is.read( (char*)&val, sizeof( val ));
    return (uint16_t)((val>8) | (val<8));
}

uint32_t readint32(std::istream &is )
{
    uint32_t val;
    is.read( (char*)&val, sizeof( val ));
    val = ((val<8)&0xFF00FF00) | ((val>8)&0x00FF00FF);
    return (val>16) | (val<16);
}

float readfloat32(std::istream &is )
{
    union { uint32_t val; float f; } dual;
    dual.val = readint32(is);
    return dual.f;
}

struct el
{
    uint16_t v1;
    uint16_t v2;
    uint16_t v3;
    void read( std::istream & is )
    {
        v1 = readint16(is);
        v2 = readint16(is);
        v3 = readint16(is);
    }
    void print( std::ostream & os )
    {
        os << std::hex << v1 << "\t" << v2 << "\t" << v3 << std::endl;
    }
}

```

```

};

std::vector<el> Vel;

void readelement( std::istream & is )
{
    el e;
    e.read( is );
    Vel.push_back( e );
}

void printascii( uint16_t tag, const char *buffer, size_t len )
{
    std::ostream & os = std::cout;
    if( tag == 0x72 )
    {
        os << "\n ";
        for( size_t i = 0; i < len; ++i )
        {
            const char &c = buffer[i];
            if( c == 0x0 ) os << " ";
            else if( c == 0x0f ) os << " ";
            else if( c == 0x17 ) os << " ";
            else if( c == 0x14 ) os << " ";
            else if( c == 0x10 ) os << " ";
            else if( c == 0x16 ) os << " ";
            else if( c == 0x08 ) os << " ";
            else if( c == 0x0b ) os << " ";
            else if( c == 0x0e ) os << " ";
            else if( c == 0x07 ) os << " ";
            else os << c;
        }
        os << " ";
    }
    else
    {
        (void)len;
        os << " " << buffer << " ";
    }
}

bool DumpADAC( std::istream & is )
{
    std::ostream &os = std::cout;

    char magic[6 + 1];
    magic[6] = 0;
    is.read( magic, 6);
    // std::cout << magic << " ";
    gdcmm_assert( strcmp( magic, "adac01" ) == 0 );
    int c = is.get();
    gdcmm_assert( c == 0 ); (void)c;
    c = is.get();
    gdcmm_assert( c == 'X' );

    uint16_t v;
    v = readint16(is);
    // std::cout << v << std::endl;
    gdcmm_assert( v == 512 ); (void)v; // ??

    int nel = 87;
    for (int i = 0; i <= nel; ++i )
    {
        readelement( is );
    }

    char buffer[512];
    for( int i = 0; i <= nel; ++i )
    {
        const el &e = Vel[i];
        int diff;
        if( i == nel )
        {
            diff = 2048 - e.v3;
            if( diff > 512 ) diff = 512;
        }
        else
        {
            const el &enext = Vel[i+1];
            diff = enext.v3 - e.v3;
        }
    }
}

```

```

is.seekg( e.v3, std::ios::beg );
//std::cout << "(" << std::hex << std::setw( 2 ) << std::setfill( '0' ) << e.v1 << ")" << std::hex << std::setw( 3 ) << std::setfill( '0' ) << e.v2 << " ";
printname( diff, 0, e.v1 );
int mult = 1;
if( e.v2 == 0 )
{
    is.read( buffer, diff);
    buffer[ diff ] = 0;
    printascii( e.v1, buffer, diff);
}
else if( e.v2 == 0x100 )
{
    mult = diff / 2;
    gdcmm_assert( diff == 2 * mult );
    for ( int ii = 0; ii < mult; ++ii )
    {
        if ( ii ) os << "\\ ";
        uint16_t val = readint16(is);
        os << " " << std::dec << val << " ";
    }
}
else if( e.v2 == 0x200 )
{
    gdcmm_assert( diff == 4 );
    uint32_t val = readint32(is);
    os << " " << std::dec << val << " ";
}
else if( e.v2 == 0x300 )
{
    gdcmm_assert( diff == 4 );
    float val = readfloat32(is);
    os << " " << std::dec << val << " ";
}
else
{
    gdcmm_assert( 0 );
}
os << std::endl;
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

    // (0019,1061) UN (OB) 61\64\61\63\30 # 2048,1 Ver200 ADAC Pegasys Headers
    const gdcmm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
    const gdcmm::DataElement& ver200adacpegasysheaders = ds.GetDataElement( tver200adacpegasysheaders );
    if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
    const gdcmm::ByteValue * bv = ver200adacpegasysheaders.GetByteValue();

    // (0019,1021) US 1 # 2,1 Ver200 Number of ADAC Headers
    // TODO

    // (0019,1041) IS [2048\221184 ] # 12,1-n Ver200 ADAC Header/Image Size
    if( bv->GetLength() != 2048 ) return 1;

    gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM2> el;
    const gdcmm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
    const gdcmm::DataElement& ver200adacheaderimagesize = ds.GetDataElement( tver200adacheaderimagesize );
    el.SetFromDataElement( ver200adacheaderimagesize );
    if( el.GetValue(0) != 2048 ) return 1;

    std::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );

```

```

if( !b ) return 1;

return 0;
}

```

14.51 DumpExamCard.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*

Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66]          # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.2010032409482568018] # 56,1 Media Storage SOP Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1]      # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1]                      # 12,1 Implementation Version Name

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Triplett, William T for bringing to your attention on this ExamCard stuff
*/
#include "gdcmReader.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmBase64.h"

#include <iomanip>

static bool compfn(const char *s1, const char *s2)
{
    return strcmp(s1,s2) < 0 ? true : false;
}

static const char *PDFStrings[] = { // Keep me ordered please
    "COILSTATE", // series of string ?
    "HARDWARE_CONFIG", // series of number ?
    "PDF_CONTROL_GEN_PARS",
    "PDF_CONTROL_PREP_PARS",
    "PDF_CONTROL_RECON_PARS",
    "PDF_CONTROL_SCAN_PARS",
    "PDF_EXAM_PARS",
    "PDF_HARDWARE_PARS",
    "PDF_PREP_PARS",
    "PDF_PRESCAN_COIL_PARS",
    "PDF_SPT_PARS",
};

static bool isvalidpdfstring( const char *pdfstring )
{
    gdcm_assert( pdfstring );
    static const size_t n = sizeof( PDFStrings ) / sizeof( *PDFStrings );
    static const char **begin = PDFStrings;
    static const char **end = begin + n;
    return std::binary_search(begin, end, pdfstring, compfn);
}

```

```

}

typedef enum
{
    param_float = 0,
    param_integer = 1, // 1 « 0
    param_string = 2, // 1 « 1
    param_3, // ??
    param_enum = 4 // 1 « 2
} param_type;

static const char *gettypenamefromtype( int i)
{
    const char *ret = nullptr;
    param_type e = (param_type)i;
    switch( e )
    {
        case param_float:
            ret = "float";
            break;
        case param_integer:
            ret = "int";
            break;
        case param_string:
            ret = "string";
            break;
        case param_3:
            ret = "??";
            break;
        case param_enum:
            ret = "enum";
            break;
    }
    gdcmm_assert( ret );
    return ret;
}

struct header
{
    /*
    * TODO:
    * Looks as if we could read all int*, float* and string* at once...
    */
    int32_t v1; // offset to int pointer array ?
    uint16_t nints; // number of ints (max number?)
    uint16_t v3; // always 0 ?
    int32_t v4; // offset to float pointer array ?
    uint32_t nfloats;
    int32_t v6; // offset to string pointer array ?
    uint32_t nstrings;
    int32_t v8; // always 8 ??
    uint32_t numparams;
    uint32_t getnints() const { return nints; }
    uint32_t getnfloats() const { return nfloats; }
    uint32_t getnstrings() const { return nstrings; }
    uint32_t getnparams() const { return numparams; }
    void read( std::istream & is )
    {
        is.read( (char*)&v1, sizeof(v1));
        if( v1 == 0x01 ) {
            // direct (FIXME how should we detect this, much like TIFF ???)
            nints = 0;
            v3 = 0;
            v4 = 0;
            nfloats = 0;
            v6 = 0;
            nstrings = 0;
            v8 = 0;
            numparams = 0;
            uint32_t bla;
            is.read( (char*)&bla, sizeof(bla) );
            gdcmm_assert( bla == 0x2 || bla == 0x3 || bla == 0x4 || bla == 0x7 );
            nstrings = 1;
            numparams = 1;
        } else {
            // indirect
            is.read( (char*)&nints, sizeof(nints));
            is.read( (char*)&v3, sizeof(v3));
            gdcmm_assert( v3 == 0 ); // looks like this is always 0
            is.read( (char*)&v4, sizeof(v4));
            is.read( (char*)&nfloats, sizeof(nfloats));

```

```

        is.read( (char*)&v6,sizeof(v6));
        is.read( (char*)&nstrings,sizeof(nstrings));
        is.read( (char*)&v8,sizeof(v8));
        gdcmm_assert( v8 == 8 );
        is.read( (char*)&numparams,sizeof(numparams));
    }
}

void print( std::ostream & os )
{
    os << v1 << ", ";
    os << nints << ", ";
    os << v3 << ", ";
    os << v4 << ", ";
    os << nfloats << ", ";
    os << v6 << ", ";
    os << nstrings << ", ";
    os << v8 << ", ";
    os << numparams << std::endl;
}

};

struct param
{
    char name[32+1];
    uint8_t boolean;
    int32_t type;
    uint32_t dim;
    union {
        uint32_t val;
        char * ptr; } v4;
    int32_t /*std::streamoff*/ offset;
    param_type gettype() const { return (param_type)type; }
    uint32_t getdim() const { return dim; }
    void read_direct_int( std::istream & is ) {
        uint32_t bla;
        int max = 9;
        std::vector<uint32_t> v;
        for( int i = 0; i < max; ++i ) {
            is.read( (char*)&bla, sizeof(bla) );
            v.push_back( bla );
        }
        is.read( (char*)&bla, sizeof(bla) );
        char name0[32];
        memset(name0,0,sizeof(name0));
        gdcmm_assert( bla < sizeof(name0) );
        is.read( name0, bla );
        size_t l = strlen(name0);
        gdcmm_assert( l == bla ); (void)l;
        char * ptr = strdup( name0 );
        v4.ptr = ptr;
        type = param_string;
        dim = 1;
        offset = 0; // important !
    }
    void read_direct_string( std::istream & is ) {
        uint32_t bla;
        is.read( (char*)&bla, sizeof(bla) );
        char name0[32*2];
        memset(name0,0,sizeof(name0));
        gdcmm_assert( bla < sizeof(name0) );
        is.read( name0, bla );
        size_t l = strlen(name0);
        gdcmm_assert( l == bla ); (void)l;
        memcpy( this->name, name0, bla );
        is.read( (char*)&bla, sizeof(bla) );
        gdcmm_assert( bla == 0x1 );
        is.read( (char*)&bla, sizeof(bla) );
        char value[32];
        memset(value,0,sizeof(value));
        gdcmm_assert( bla < sizeof(value) );
        is.read( value, bla );
        is.read( (char*)&bla, sizeof(bla) );
        gdcmm_assert( bla == 0 ); // trailing stuff ?
        is.read( (char*)&bla, sizeof(bla) );
        gdcmm_assert( bla == 0 ); // trailing stuff ?
        const uint32_t cur = (uint32_t)is.tellg();
        std::cerr << "offset:" << cur << std::endl;
        if( cur == 65 )
            is.read( (char*)&bla, 1 );
        else if( cur == 66 )
            is.read( (char*)&bla, 1 );
    }
}

```

```

else if( cur == 89 )
    is.read( (char*)&bla, 1 );
else if( cur == 95 )
    is.read( (char*)&bla, 2 );
else if( cur == 122 )
    is.read( (char*)&bla, 2 );
else
    gdcmm_assert(0);
type = param_string;
dim = 1;
// FIXME: store the value in v4 for now:
char * ptr = strdup( value );
v4.ptr = ptr;
offset = 0; // important !
}
void read( std::istream & is )
{
    is.read( name, 32 + 1);
    // This is always the same issue the string can contains garbage from previous run,
    // we need to print only until the first \0 character:
    gdcmm_assert( strlen( name ) <= 32 );
    is.read( (char*)&boolean,1);
    gdcmm_assert( boolean == 0 || boolean == 1 || boolean == 0x69 || boolean == 128 || boolean == 95 || boolean == 116 ); // some
        kind of bool, or digital trash ?
    is.read( (char*)&type, sizeof( type ) );
    gdcmm_assert( gettypenameefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) ); // number of elements
    is.read( (char*)&v4.val, sizeof( v4.val ) );
    //gdcmm_assert( v4.val == 0 ); // always 0 ? sometimes not...
    const uint32_t cur = (uint32_t)is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    gdcmm_assert( offset != 0 );
    offset += cur;
}

void print( std::ostream & os ) const
{
    os << name << ", ";
    os << (int)boolean << ", ";
    os << type << ", ";
    os << dim << ", ";
    os << v4.val << ", ";
    os << offset << std::endl;
}

void printvalue( std::ostream & os, std::istream & is ) const
{
    if( offset ) {
        is.seekg( offset );
        switch( type )
        {
            case param_float:
            {
                os.precision(2);
                os << std::fixed;
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ", ";
                    float v;
                    is.read( (char*)&v, sizeof(v) );
                    os << v; // what if the string contains \0 ?
                }
            }
            break;
            case param_integer:
            {
                int32_t v;
                for( uint32_t idx = 0; idx < dim; ++idx )
                {
                    if( idx ) os << ", ";
                    is.read( (char*)&v, sizeof(v) );
                    os << v;
                }
            }
            break;
            case param_string:
            {
                int size = 81;
                std::string v;
                v.resize( size );
                for( uint32_t idx = 0; idx < dim; ++idx )
                {

```

```

        if( idx ) os << " ";
        is.read( &v[0], size );
        os << v.c_str();
    }
    break;
case param_enum:
{
    int32_t v;
    for( uint32_t idx = 0; idx < dim; ++idx )
    {
        if( idx ) os << " ";
        is.read( (char*)&v, sizeof(v) );
        os << v;
    }
    break;
}
} else {
#ifdef 1
    // direct
    assert ( type == param_string );
    char * ptr = v4.ptr;
    //std::string v;
    //v.resize( dim );
    //is.read( &v[0], dim );
    os << ptr;
#endif
}

}

void printxml( std::ostream & os, std::istream & is ) const
{
    // <Attribute Name="CGEN_force_par_mode" Type="enum">0</Attribute>
    os << " <Attribute";
    os << " Name=\"" << name << "\"";
    os << " Type=\"" << gettypenamefromtype(type) << "\"";
    if( dim != 1 )
    {
        os << " ArraySize=\"" << dim << "\"";
    }
    os << ">";
    printvalue( os, is );
    os << "</Attribute>\n";
}

void printcsv( std::ostream & os, std::istream & is ) const
{
    os << std::setw(32) << std::left << name << " ";
    os << std::setw(7) << std::right << gettypenamefromtype(type) << " ";
    os << std::setw(4) << dim << " ";
    os << " ";
    printvalue( os, is );
    os << ",\n";
}
};

static bool ProcessNested( gdcmm::DataSet & ds )
{
    /*
    TODO:
    Looks like the real length of the blob is stored here:
    (2005,1132) SQ # u/1,1 ?
    (ffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ] # 26,1 Private Creator
    (2005,1143) SL 3103 # 4,1 ?

    Wotsit ?
    (2005,1132) SQ # u/1,1 ?
    (ffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ] # 26,1 Private Creator
    (2005,1147) CS [Y ] # 2,1 ?
    */
    bool ret = false;

    // (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS] # 20,1 Protocol Data Name
    const gdcmm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt0 ) ) return false;
    const gdcmm::DataElement &de0 = ds.GetDataElement( pt0 );
    if( de0.IsEmpty() ) return false;
    const gdcmm::ByteValue * bv0 = de0.GetByteValue();

```



```

std::string s0( bv0->GetPointer() , bv0->GetLength() );

// (2005,1139) LO [IEEE_PDF] # 8,1 Protocol Data Type
const gdcmm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
if( !ds.FindDataElement( pt1 ) ) return false;
const gdcmm::DataElement &de1 = ds.GetDataElement( pt1 );

// (2005,1143) SL 53 # 4,1 Protocol Data Block Length (non-padded)
const gdcmm::PrivateTag pt2(0x2005,0x43,"Philips MR Imaging DD 002");
if( !ds.FindDataElement( pt2 ) ) return false;
const gdcmm::DataElement &de2 = ds.GetDataElement( pt2 );

// (2005,1147) CS [Y] # 2,1 Protocol Data Boolean
const gdcmm::PrivateTag pt3(0x2005,0x47,"Philips MR Imaging DD 002");
if( !ds.FindDataElement( pt3 ) ) return false;
const gdcmm::DataElement &de3 = ds.GetDataElement( pt3 );
(void)de3;

// (2005,1144) OW 00\00\00\00\05\00\00\00\35\2e\31\2e\37\00 # 54,1 Protocol Data Block
const gdcmm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
if( !ds.FindDataElement( pt ) ) return false;
const gdcmm::DataElement &de = ds.GetDataElement( pt );
if( de.IsEmpty() ) return false;
const gdcmm::ByteValue * bv = de.GetByteValue();

if( s0 == "ExamCardBlob" )
{
    gdcmm_assert( de1.IsEmpty() );

    std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".xml";
    std::ofstream out( fn.c_str() );

    // remove trailing \0
    size_t len = strlen( bv->GetPointer() );
    out.write( bv->GetPointer() , len );
    out.close();

    // Extract binary64 thingy (this is a ugly hack, better use an XML parser)
    std::string dup( bv->GetPointer(), len );
    std::string::size_type pos1 = dup.find( "<ExamCardBlob>" );
    std::string::size_type pos2 = dup.find( "</ExamCardBlob>" );

    std::string b64( bv->GetPointer() + pos1 + 14, pos2 - (pos1 + 14) );

    // ulgy hack to remove \r\n from input base64:
    std::string::iterator r_pos = std::remove(b64.begin(), b64.end(), '\r');
    b64.erase(r_pos, b64.end());
    std::string::iterator n_pos = std::remove(b64.begin(), b64.end(), '\n');
    b64.erase(n_pos, b64.end());
#ifdef 0
    std::ofstream out2( "debug" );
    out2.write( b64.c_str(), b64.size() );
    out2.close();
#endif

    const size_t dlen = gdcmm::Base64::GetDecodeLength(b64.c_str(), b64.size() );

    std::string decoded;
    decoded.resize( dlen );
    gdcmm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );

    std::ofstream f64( "soap.xml" );
    f64.write( decoded.c_str(), decoded.size() );
    f64.close();

    ret = true;
}
else
{
    if( de1.IsEmpty() ) return false;
    const gdcmm::ByteValue * bv1 = de1.GetByteValue();
    gdcmm::Element<gdcmm::VR::SL,gdcmm::VM::VM1> dlen = {{0L}};
    dlen.SetFromDataElement( de2 );
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
    {
        std::istringstream is;
        gdcmm_assert( bv->GetLength() == (size_t)dlen.GetValue() || bv->GetLength() == (size_t)(dlen.GetValue() + 1) );
    }
}

```

```

std::string dup( bv->GetPointer(), dlen.GetValue() /*bv->GetLength()*/ );
is.str( dup );

header h;
h.read( is );
//gdcmm_assert( is.peek() && is.eof() );
#ifdef 1
static int c = 0;
std::string fn0 = gdcmm::LOComp::Trim( s1.c_str() ); // remove trailing space
std::stringstream ss;
ss << fn0 << " " << c++;
if( h.v1 == 0x01 )
    ss << ".direct";
else
    ss << ".indirect";
std::cout << "fn0=" << ss.str() << " Len=" << bv->GetLength() << std::endl;
std::ofstream out( ss.str().c_str() );
out.write( bv->GetPointer(), bv->GetLength() );
out.close();
#endif
#ifdef 1
std::cout << dup.c_str() << std::endl;
h.print( std::cout );
#endif

std::vector< param > params;
if( h.v1 == 0x01 ) {
    for( uint32_t i = 0; i < 1 /* h.getnparams()*/; ++i ) {
        param p;
        if( s0 == "HARDWARE_CONFIG " )
        {
            p.read_direct_int( is );
        }
        else if( s0 == "COILSTATE " )
        {
            p.read_direct_string( is );
        }
        else
        {
            gdcmm_assert(0);
        }
        params.push_back( p );
    }
} else {
    gdcmm_assert( is.tellg() == std::streampos(0x20) );
    is.seekg( 0x20 );

    param p;
    for( uint32_t i = 0; i < h.getnparams(); ++i )
    {
        p.read( is );
        //p.print( std::cout );
        params.push_back( p );
    }
}

std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
bool b1 = isvalidpdfstring( fn.c_str() );
gdcmm_assert( b1 ); (void)b1;
fn += ".csv";
//fn += ".xml";
std::ofstream csv( fn.c_str() );

// let's do some bookkeeping:
uint32_t nfloats = 0;
uint32_t nints = 0;
uint32_t nstrings = 0;
for( std::vector<param>::const_iterator it = params.begin();
    it != params.end(); ++it )
{
    param_type type = it->gettype();
    switch( type )
    {
        case param_float:
            nfloats += it->getdim();
            break;
        case param_integer:
            nints += it->getdim();
            break;
        case param_string:
            nstrings += it->getdim();

```

```

        break;
    default:
        ;
    }
}
#endif 0
std::cout << "Stats:" << std::endl;
std::cout << "nfloats:" << nfloats << std::endl;
std::cout << "nints:" << nints << std::endl;
std::cout << "nstrings:" << nstrings << std::endl;
#endif
gdcmm_assert( h.getnints() >= nints );
gdcmm_assert( h.getnfloats() >= nfloats );
gdcmm_assert( h.getnstrings() >= nstrings );

for( uint32_t i = 0; i < h.getnparams(); ++i )
{
    params[i].printcsv( csv, is );
    //params[i].printxml( csv, is );
}
csv.close();
ret = true;
}
else if( s1 == "ASCII " )
{
#endif 0
std::cerr << "ASCII is not handled" << std::endl;
std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
fn += ".asc";
std::ofstream out( fn.c_str() );
out.write( bv->GetPointer() , bv->GetLength() );
out.close();
#endif
std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
fn += ".sin";
std::ofstream sin( fn.c_str() );

const char *beg = bv->GetPointer();
const char *end = beg + bv->GetLength();
gdcmm_assert( *beg == 0 );
const char *p = beg + 1; // skip first \0
size_t prev = 0;
for( ; p != end; ++p )
{
    if( *p == 0 )
    {
        const char *s = beg + prev + 1;
        if( *s )
        {
            sin << s << std::endl;
        }
        else
        {
            sin << std::endl;
        }
        prev = p - beg;
    }
}
sin.close();

ret = true;
}
else if( s1 == "BINARY" )
{
std::cerr << "BINARY is not handled" << std::endl;
std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
fn += ".bin";
std::ofstream out( fn.c_str() );
//out.write( bv->GetPointer() + 512, bv->GetLength() - 512 );
out.write( bv->GetPointer() , bv->GetLength() );
out.close();

#endif 0
int array[ 128 ];
memcpy( array, bv->GetPointer(), 512 );
for( int i = 0; i < 14; ++i )
{
    std::cout << array[i] << std::endl;
}

```



```

#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

bool PrintNameValueMapping( gdcm::SequenceOfItems *sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    using namespace gdcm;
    // prepare names mapping:
    typedef VRToType<VR::UL>::Type UL;
    std::map< UL, std::string > names;
    gdcm_assert( sqi_names );
    gdcm_assert( sqi_values );
    SequenceOfItems::SizeType s = sqi_names->GetNumberOfItems();
    PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
    // First sequence contains all possible names (this is a dict)
    for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
        const Item & item = sqi_names->GetItem( i );
        const DataSet & ds = item.GetNestedDataSet();
        if( !ds.FindDataElement( tindex )
            || !ds.FindDataElement( tname ) )
        {
            gdcm_assert( 0 );
            return false;
        }
        const DataElement & index = ds.GetDataElement( tindex );
        const DataElement & name = ds.GetDataElement( tname );
        if( index.IsEmpty() || name.IsEmpty() )
        {
            gdcm_assert( 0 );
            return false;
        }
        gdcm::Element<VR::UL, VM::VM1> el1;
        el1.SetFromDataElement( index );

        gdcm::Element<VR::LO, VM::VM1> el2;
        el2.SetFromDataElement( name );
        // std::cout << el1.GetValue() << " " << el2.GetValue() << std::endl;
        names.insert( std::make_pair( el1.GetValue(), el2.GetValue() ) );
    }

    SequenceOfItems::SizeType s2 = sqi_values->GetNumberOfItems();
    gdcm_assert( s2 <= s );
    PrivateTag tindex2(0x7fe1,0x48,"GEMS_Ultrasound_MovieGroup_001");
    for( SequenceOfItems::SizeType i = 1; i <= s2; ++i )
    {
        const Item & item = sqi_values->GetItem( i );
        const DataSet & ds = item.GetNestedDataSet();
        if( !ds.FindDataElement( tindex2 ) )
        {
            gdcm_assert( 0 );
            return false;
        }
        const DataElement & index2 = ds.GetDataElement( tindex2 );
        if( index2.IsEmpty() )
        {
            gdcm_assert( 0 );
            return false;
        }
        gdcm::Element<VR::FD, VM::VM1_2> el1;
        el1.SetFromDataElement( index2 );

        UL copy = (UL)el1.GetValue();
        #if 1
            std::cout << indent;
            std::cout << "( " << names[ copy ];
        #endif
        // (7fe1,1052) FD 1560 # 8,1 ?
        // (7fe1,1057) LT [MscSkelSup] # 10,1 ?
        //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
        PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"); // UL
        PrivateTag tvaluefloat1(0x7fe1,0x51,"GEMS_Ultrasound_MovieGroup_001"); // FL
    }
}

```

```

PrivateTag tvaluefloat(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001"); // FD
PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"); // SL
PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"); // OB
PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001"); // LT
PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"); // SL / 1-N
PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"); // SL ??
PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001"); // FD ??
#endif
// if 1
std::cout << " ) = ";
#endif
if( ds.FindDataElement( tvalueint ) )
{
    const DataElement & value = ds.GetDataElement( tvalueint );
    gdcm::Element<VR::UL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluefloat1 ) )
{
    const DataElement & value = ds.GetDataElement( tvaluefloat1 );
    gdcm::Element<VR::FL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluefloat ) )
{
    const DataElement & value = ds.GetDataElement( tvaluefloat );
    gdcm::Element<VR::FD,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluesl ) )
{
    const DataElement & value = ds.GetDataElement( tvaluesl );
    gdcm::Element<VR::SL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvalueul ) )
{
    const DataElement & value = ds.GetDataElement( tvalueul );
    gdcm::Element<VR::UL,VM::VM1_n> el2;
    el2.SetFromDataElement( value );
    gdcm_assert( el2.GetLength() == 1 );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvalueob ) )
{
    const DataElement & value = ds.GetDataElement( tvalueob );
    // gdcm::Element<VR::SL,VM::VM1> el2;
    // el2.SetFromDataElement( value );
    // std::cout << el2.GetValue() << std::endl;
    std::cout << value << std::endl;
}
else if( ds.FindDataElement( tvaluetext ) )
{
    const DataElement & value = ds.GetDataElement( tvaluetext );
    gdcm::Element<VR::LT,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluesl2 ) )
{
    const DataElement & value = ds.GetDataElement( tvaluesl2 );
    gdcm::Element<VR::SL,VM::VM1_n> el2;
    el2.SetFromDataElement( value );
    el2.Print( std::cout );
    gdcm_assert( el2.GetLength() == 4 );
    std::cout << std::endl;
}
else if( ds.FindDataElement( tvaluesl3 ) )
{
    const DataElement & value = ds.GetDataElement( tvaluesl3 );
    gdcm::Element<VR::SL,VM::VM1_n> el2;
    el2.SetFromDataElement( value );
    el2.Print( std::cout );
    // gdcm_assert( el2.GetLength() == 4 );
    std::cout << std::endl;
}

```

```

    }
    else if( ds.FindDataElement( tvaluefd ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefd );
        gdcm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        // gdcm_assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength() == 8 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefloat2 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefloat2 );
        gdcm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        gdcm_assert( el2.GetLength() == 2 );
        std::cout << std::endl;
    }
    else if( ds.FindDataElement( tvaluefd1 ) )
    {
        const DataElement & value = ds.GetDataElement( tvaluefd1 );
        gdcm::Element<VR::FD,VM::VM1_n> el2;
        el2.SetFromDataElement( value );
        el2.Print( std::cout );
        gdcm_assert( el2.GetLength() == 4 );
        std::cout << std::endl;
    }
    else
    {
        std::cout << "(no value)" << std::endl;
        // std::cout << ds << std::endl;
        gdcm_assert( ds.Size() == 2 );
    }
}
return true;
}

bool PrintNameValueMapping2( gdcm::PrivateTag const & privtag, const gdcm::DataSet & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag ) ) return false;
    const gdcm::DataElement& seq_values = ds.GetDataElement( privtag );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = seq_values.GetValueAsSQ();

    return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1, gdcm::PrivateTag const & privtag2, const gdcm::DataSet & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag1 ) )
    {
        gdcm_assert( 0 );
        return false;
    }
    const gdcm::DataElement& values10name = ds.GetDataElement( privtag1 );
    gdcm::Element<gdcm::VR::LO,gdcm::VM::VM1> el;
    el.SetFromDataElement( values10name );
    std::cout << std::endl;
    std::cout << " <" << el.GetValue().c_str() << ">" << std::endl;

    return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}

bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    const gdcm::PrivateTag tseq_values73(0x7fe1,0x73,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values73 ) )
    {
        std::cout << indent << "No group 73" << std::endl;
        return false;
    }
    const gdcm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values73 = seq_values73.GetValueAsSQ();

    size_t ni3 = sqi_values73->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcm::Item &item_73 = sqi_values73->GetItem(i3);
        gdcm::DataSet &ds73 = item_73.GetNestedDataSet();

```

```

    gdcmm_assert( ds73.Size() == 3 );

    const gdcmm::PrivateTag tseq_values74name(0x7fe1,0x74,"GEMS_Ultrasound_MovieGroup_001");
    const gdcmm::PrivateTag tseq_values75(0x7fe1,0x75,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict, indent);
    std::cout << std::endl;
}
return true;
}

bool print36( gdcmm::DataSet const & ds10, gdcmm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    (void)sqi_dict;
    const gdcmm::PrivateTag tseq_values36(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values36 ) )
    {
        std::cout << indent << "No group 36" << std::endl;
        return false;
    }
    const gdcmm::DataElement& seq_values36 = ds10.GetDataElement( tseq_values36 );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values36 = seq_values36.GetValueAsSQ();

    size_t ni3 = sqi_values36->GetNumberOfItems();
    gdcmm_assert( ni3 >= 1 );
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcmm::Item &item_36 = sqi_values36->GetItem(i3);
        gdcmm::DataSet &ds36 = item_36.GetNestedDataSet();
        gdcmm_assert( ds36.Size() == 4 );

        // (7fe1,1037) UL 47 # 4,1 US MovieGroup Number of Frames
        // (7fe1,1043) OB 40\00\1c\c4\67\2f\0b\11\40 # 376,1 ?
        // (7fe1,1060) OB 4e\4e\49\4f\4e\47\46\43\2a # 4562714,1 US MovieGroup Image Data
        //
        const gdcmm::PrivateTag timagedata(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");
        gdcmm_assert( ds36.FindDataElement( timagedata ) );
        gdcmm::DataElement const & imagedata = ds36.GetDataElement( timagedata );

        const gdcmm::ByteValue * bv = imagedata.GetByteValue();
        gdcmm_assert( bv );
        static int c = 0;
        std::stringstream ss;
        ss << "/tmp/debug";
        ss << c++;
        std::ofstream os( ss.str().c_str(), std::ios::binary );
        os.write( bv->GetPointer(), bv->GetLength() );
        os.close();

        //const gdcmm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
        //PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
        //std::cout << std::endl;
    }
    return true;
}

bool print83( gdcmm::DataSet const & ds10, gdcmm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    const gdcmm::PrivateTag tseq_values83(0x7fe1,0x83,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values83 ) )
    {
        std::cout << indent << "No group 83" << std::endl;
        return false;
    }
    const gdcmm::DataElement& seq_values83 = ds10.GetDataElement( tseq_values83 );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values83 = seq_values83.GetValueAsSQ();

    size_t ni3 = sqi_values83->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcmm::Item &item_83 = sqi_values83->GetItem(i3);
        gdcmm::DataSet &ds83 = item_83.GetNestedDataSet();
        gdcmm_assert( ds83.Size() == 3 );

        const gdcmm::PrivateTag tseq_values84name(0x7fe1,0x84,"GEMS_Ultrasound_MovieGroup_001");
        const gdcmm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
        PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
        std::cout << std::endl;
    }
    return true;
}

bool PrintNameValueMapping4( gdcmm::PrivateTag const & privtag0, const gdcmm::DataSet & subds, gdcmm::PrivateTag const &

```



```

        privtag1, gdcmm::PrivateTag const & privtag2,
gdcmm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    (void)indent;
    if( !subds.FindDataElement( privtag0 ) )
    {
        gdcmm_assert( 0 );
        return false;
    }
    const gdcmm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values10 = seq_values10.GetValueAsSQ();

    size_t ni1 = sqi_values10->GetNumberOfItems();
    // gdcmm_assert( ni1 == 1 );
    for( size_t i1 = 1; i1 <= ni1; ++i1 )
    {
        gdcmm::Item &item_10 = sqi_values10->GetItem(i1);
        gdcmm::DataSet &ds10 = item_10.GetNestedDataSet();
        gdcmm_assert( ds10.Size() == 2 + 3 );
        // (7fe1,0010)
        // (7fe1,1012)
        // (7fe1,1018)
        // (7fe1,1020)
        // (7fe1,1083)

        PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, " " );
        std::cout << std::endl;

        const gdcmm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
        if( !ds10.FindDataElement( tseq_values20 ) )
        {
            gdcmm_assert( 0 );
            return false;
        }
        const gdcmm::DataElement& seq_values20 = ds10.GetDataElement( tseq_values20 );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values20 = seq_values20.GetValueAsSQ();

        size_t ni2 = sqi_values20->GetNumberOfItems();
        //gdcmm_assert( ni == 1 );
        for( size_t i2 = 1; i2 <= ni2; ++i2 )
        {
            gdcmm::Item &item_20 = sqi_values20->GetItem(i2);
            gdcmm::DataSet &ds20 = item_20.GetNestedDataSet();
            size_t count = ds20.Size(); (void)count;
            gdcmm_assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
            // (7fe1,0010)
            // (7fe1,1024)
            // (7fe1,1026)
            // (7fe1,1036)
            // (7fe1,103a)
            // (7fe1,1083) (*)

            const gdcmm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001");
            const gdcmm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
            PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, " " );
            std::cout << std::endl;

            print36(ds20, sqi_dict, " ");
            print83(ds20, sqi_dict, " ");
        }

        print83(ds10, sqi_dict, " ");
    }
    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcmm;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;

```

```

const DataElement& seq = ds.GetDataElement( tseq );

SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
gdcm_assert( sqi->GetNumberOfItems() == 1 );

Item &item = sqi->GetItem(1);
DataSet &subds = item.GetNestedDataSet();

const PrivateTag tseq_dict(0x7fe1,0x70,"GEMS_Ultrasound_MovieGroup_001");
if( !subds.FindDataElement( tseq_dict ) ) return 1;
const DataElement& seq_dict = subds.GetDataElement( tseq_dict );
SmartPointer<SequenceOfItems> sqi_dict = seq_dict.GetValueAsSQ();

const PrivateTag tseq_values8(0x7fe1,0x8,"GEMS_Ultrasound_MovieGroup_001");
if( !subds.FindDataElement( tseq_values8 ) ) return 1;
const DataElement& seq_values8 = subds.GetDataElement( tseq_values8 );
SmartPointer<SequenceOfItems> sqi_values8 = seq_values8.GetValueAsSQ();

const PrivateTag tseq_values8name(0x7fe1,0x2,"GEMS_Ultrasound_MovieGroup_001");
if( !subds.FindDataElement( tseq_values8name ) ) return 1;
const DataElement& values8name = subds.GetDataElement( tseq_values8name );
{
    Element<VR::LO,VM::VM1> el;
    el.SetFromDataElement( values8name );
    std::cout << el.GetValue() << std::endl;
}
size_t count = subds.Size(); (void)count;
gdcm_assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2);

// (7fe1,0010) # 30,1 Private Creator
// (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
// (7fe1,1003) # 4,1 ?
// (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
// (7fe1,1010) # 1372196,1 ?
// (7fe1,1070) # 33684,1 US MovieGroup Dict
// (7fe1,1073) (*)
PrintNameValueMapping( sqi_values8, sqi_dict, " ");

const PrivateTag tseq_values10(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values10name(0x7fe1,0x12,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq_values18(0x7fe1,0x18,"GEMS_Ultrasound_MovieGroup_001");
PrintNameValueMapping4( tseq_values10, subds, tseq_values10name, tseq_values18, sqi_dict, " ");

print73( subds, sqi_dict, " ");

#if 0
gdcm::DataSet::ConstIterator it = subds.Begin();
for( ; it != subds.End(); ++it )
{
    const gdcm::DataElement &de = *it;
    std::cout << de.GetTag() << std::endl;
}
#endif

return 0;
}

```

14.53 DumpImageHeaderInfo.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Dump TOSHIBA MDW HEADER / Image Header Info
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"

```

Generated by Doxygen

```

    os << " " << val << std::endl;
    p += sizeof(val);
    char str2[17];
    memcpy( str2, p, 16 );
    str2[16] = 0;
    os << " " << str2 << std::endl;
}

#ifdef 0
std::ofstream out( str, std::ios::binary );
out.write( (char*)&magic, sizeof( magic ) );
out.write( (char*)&l, sizeof( l ) );
out.write( str, 16 );
out.write( &bytes[0], bytes.size() );
#endif
return is;
}

static bool DumpImageHeaderInfo( std::istream & is, size_t refln )
{
    // TUSNONIMAGESTAM (5176)
    // TUSREMEASUREMEN (1352)
    // TUSBSINGLELAYOU (16)
    // TUSCLIPPARETE (104)

    element el;
    while( el.read( is ) )
    {
        //size_t pos = is.tellg();
        //gdcmm_assert( pos == refln );
        (void)refln;

        return true;
    }
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

    const gdcmm::PrivateTag timageheaderinfo(0x0029,0x10,"TOSHIBA MDW HEADER");
    if( !ds.FindDataElement( timageheaderinfo ) ) return 1;
    const gdcmm::DataElement& imageheaderinfo = ds.GetDataElement( timageheaderinfo );
    if ( imageheaderinfo.IsEmpty() ) return 1;
    const gdcmm::ByteValue *bv = imageheaderinfo.GetByteValue();

    std::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpImageHeaderInfo( is, bv->GetLength() );
    if( !b ) return 1;

#ifdef 0
    const float d1 = 0.0041666668839752674; // 89 88 88 3B // 0x44c
    //const float d1 = 0.053231674455417881;
    const float d2 = 0.10828025639057159; // 0A C2 DD 3D // 0x1ac
    //const float d1 = 0.17869562069272813;
    //const unsigned int d2 = 4294967280;
    const float d3 = 0.10828025639057159; // 0A C2 DD 3D // 0x15c
    const int32_t d4 = 134;
    const uint32_t d5 = 1153476;
    std::ofstream t("/tmp/debug", std::ios::binary );
    //t.write( (char*)&d0, sizeof( d0 ) );
    t.write( (char*)&d1, sizeof( d1 ) );
    t.write( (char*)&d2, sizeof( d2 ) );
    t.write( (char*)&d3, sizeof( d3 ) );
    t.write( (char*)&d4, sizeof( d4 ) );
    t.write( (char*)&d5, sizeof( d5 ) );
    t.close();
#endif

    return 0;
}

```

}

14.54 DumpPhilipsECHO.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmDeflateStream.h"
#include "gdcm_zlib.h"

/*
 * This example extract the ZLIB compressed US image from a Philips private tag
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Usage:
 *
 * $ DumpPhilipsECHO private_us.dcm raw_us_img.raw
 * $ gdcming --sop-class-uid 1.2.840.10008.5.1.4.1.1.3.1 --size 608,427,88 raw_us_img.raw raw_us_img.dcm
 */

// header:
struct hframe
{
    uint32_t val0; // 800 increment ?
    uint16_t val1[2];
    uint16_t val2[2];
    uint32_t imgsize;

    bool operator==(const hframe &h) const
    {
        return val0 == h.val0 &&
            val1[0] == h.val1[0] &&
            val1[1] == h.val1[1] &&
            val2[0] == h.val2[0] &&
            val2[1] == h.val2[1] &&
            imgsize == h.imgsize;
    }
};

static bool ProcessDeflate( const char *outfilename, const int nslices, const
    int buf_size, const char *buf, const std::streampos len,
    const char *crcbuf, const size_t crclen )
{
    std::vector< hframe > crcheaders;
    crcheaders.reserve( nslices );
    {
        std::istream is;
        is.str( std::string( crcbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ));
        }
    }
    #if 0
        std::cout << header.val0
            << " " << header.val1[0]
            << " " << header.val1[1]
            << " " << header.val2[0]

```

```

        « " " « header.val2[1]
        « " " « header.imgsize « std::endl;
#endif
    crchheaders.push_back( header );
}

std::istream is;
is.str( std::string( buf, (size_t)len ) );

std::streamoff totalsize;
is.read( (char*)&totalsize, sizeof( totalsize ));
gdc_m_assert( totalsize == len );

uint32_t nframes;
is.read( (char*)&nframes, sizeof( nframes ));
gdc_m_assert( nframes == (uint32_t)nslices );

std::vector< std::streamoff > offsets;
offsets.reserve( nframes );
for( uint32_t frame = 0; frame < nframes ; ++frame )
{
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
}

std::vector<char> outbuf;

const int size[2] = { 608, 427 }; // FIXME: where does it comes from ?
std::stringstream ss;
ss << outfilename;
ss << ' ';
//ss << crchheaders[0].imgsize; // FIXME: Assume all header are identical !
ss << size[0];
ss << ' ';
ss << size[1];
ss << ' ';
ss << nframes;
ss << ".raw";
std::ofstream os( ss.str().c_str(), std::ios::binary );

gdc_m_assert( buf_size >= size[0] * size[1] );
outbuf.resize( buf_size );

hframe header;
//uint32_t prev = 0;
for( unsigned int r = 0; r < nframes; ++r )
{
    is.read( (char*)&header, sizeof( header ));

    gdc_m_assert( header == crchheaders[r] );
    gdc_m_assert( header.val1[0] == 2000 );
    gdc_m_assert( header.val1[1] == 3 );
    gdc_m_assert( header.val2[0] == 1 );
    gdc_m_assert( header.val2[1] == 1280 );

    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)outbuf.data();
    gdc_m_assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (const Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
        sourceLen = (uLong)totalsize - (uLong)offsets[r] - 16;
    else
        sourceLen = (uLong)offsets[r+1] - (uLong)offsets[r] - 16;
    // FIXME: in-memory decompression:
    int ret = uncompress( dest, &destLen, source, sourceLen);
    gdc_m_assert( ret == Z_OK ); (void)ret;
    gdc_m_assert( destLen >= (uLongf)size[0] * size[1] ); // 16bytes padding ?
    gdc_m_assert( header.imgsize == (uint32_t)size[0] * size[1] );
    //os.write( &outbuf[0], outbuf.size() );
    os.write( outbuf.data(), size[0] * size[1] );

    // skip data:
    is.seekg( sourceLen, std::ios::cur );
}
os.close();
gdc_m_assert( is.tellg() == totalsize );

return true;

```

```

}

static bool ProcessNone( const char *outfilename, const int nslices, const
int buf_size, const char *buf, const std::streampos len,
const char *crdbuf, const size_t crclen )
{
    std::vector< hframe > crcheaders;
    crcheaders.reserve( nslices );
    {
        std::istream is;
        is.str( std::string( crdbuf, crclen ) );
        hframe header;
        for( int r = 0; r < nslices; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
#if 0
            std::cout << header.val0
                << " " << header.val1[0]
                << " " << header.val1[1]
                << " " << header.val2[0]
                << " " << header.val2[1]
                << " " << header.imgsize << std::endl;
#endif
            crcheaders.push_back( header );
        }

        std::istream is;
        is.str( std::string( buf, (size_t)len ) );

        std::streampos totalsize;
        is.read( (char*)&totalsize, sizeof( totalsize ) );
        gdcmm_assert( totalsize == len );

        uint32_t nframes;
        is.read( (char*)&nframes, sizeof( nframes ) );
        gdcmm_assert( nframes == (uint32_t)nslices );

        std::vector< uint32_t > offsets;
        offsets.reserve( nframes );
        for( uint32_t frame = 0; frame < nframes; ++frame )
        {
            uint32_t offset;
            is.read( (char*)&offset, sizeof( offset ) );
            offsets.push_back( offset );
            //std::cout << offset << std::endl;
        }

        std::vector<char> outbuf;
        // No idea how to present the data, I'll just append everything, and present it as 2D
        std::stringstream ss;
        ss << "outfilename";
        ss << " ";
        ss << crcheaders[0].imgsize; // FIXME: Assume all header are identical !
        ss << " ";
        ss << nframes;
        ss << ".raw";
        std::ofstream os( ss.str().c_str(), std::ios::binary );
        outbuf.resize( buf_size ); // overallocated + 16
        char *buffer = outbuf.data();

        hframe header;
        for( unsigned int r = 0; r < nframes; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
#if 0
            std::cout << header.val0
                << " " << header.val1[0]
                << " " << header.val1[1]
                << " " << header.val2[0]
                << " " << header.val2[1]
                << " " << header.imgsize << std::endl;
#endif
            gdcmm_assert( header == crcheaders[r] );

            is.read( buffer, buf_size - 16 );
            os.write( buffer, header.imgsize );
        }
        gdcmm_assert( is.tellg() == totalsize );

```

```

os.close();

return true;
}

#ifdef NDEBUG
static const char * const UDM_USD_DATATYPE_STRINGS[] = {
    "UDM_USD_DATATYPE_DIN_2D_ECHO",
    "UDM_USD_DATATYPE_DIN_2D_ECHO_CONTRAST",
    "UDM_USD_DATATYPE_DIN_DOPPLER_CW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW",
    "UDM_USD_DATATYPE_DIN_DOPPLER_PW_TDI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_FLOW",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_PMI",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_CPA",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_MMODE_ECHO",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_TDI",
    "UDM_USD_DATATYPE_DIN_PARAM_BLOCK",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_POWER",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_DOPPLER_AUDIO",
    "UDM_USD_DATATYPE_DIN_DOPPLER_HIGHQ",
    "UDM_USD_DATATYPE_DIN_PHYSIO",
    "UDM_USD_DATATYPE_DIN_2D_COLOR_STRAIN",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_RGB",
    "UDM_USD_DATATYPE_DIN_XFOV_REALTIME_GRAPHICS",
    "UDM_USD_DATATYPE_DIN_XFOV_MOSAIC",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_R",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_G",
    "UDM_USD_DATATYPE_DIN_COMPOSITE_B",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VELOCITY",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_POWER",
    "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VARIANCE",
    "UDM_USD_DATATYPE_DIN_2D_ELASTO",
};

static inline bool is_valid( const char * datatype_str )
{
    static const int n = sizeof( UDM_USD_DATATYPE_STRINGS ) / sizeof( *UDM_USD_DATATYPE_STRINGS );
    bool found = false;
    if( datatype_str )
    {
        for( int i = 0; !found && i < n; ++i )
        {
            found = strcmp( datatype_str, UDM_USD_DATATYPE_STRINGS[i] ) == 0;
        }
    }
    return found;
}
#endif

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds1 = file.GetDataSet();

    const PrivateTag tseq1(0x200d,0x3cf8,"Philips US Imaging DD 045");
    if( !ds1.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = ds1.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sq1 = seq1.GetValueAsSQ();
    gdcm_assert( sq1->GetNumberOfItems() >= 1 );

    const size_t nitems = sq1->GetNumberOfItems();
    for( size_t item = 1; item < nitems; ++item )
    {
        Item &item1 = sq1->GetItem(item);
        DataSet &ds2 = item1.GetNestedDataSet();

        // (200d,300d) LO 28 UDM_USD_DATATYPE_DIN_2D_ECHO
        const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");

```



```

if( !ds2.FindDataElement( tdatatype ) ) return 1;
const DataElement& datatype = ds2.GetDataElement( tdatatype );
const ByteValue *bvdatatype = datatype.GetByteValue();
if( !bvdatatype ) return 1;

const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
if( !ds2.FindDataElement( tseq2 ) ) return 1;
const DataElement& seq2 = ds2.GetDataElement( tseq2 );

SmartPointer<SequenceOfItems> sqi2 = seq2.GetValueAsSQ();
gdcmm_assert( sqi2->GetNumberOfItems() >= 1 );

// FIXME: what if not in first Item ?
gdcmm_assert( sqi2->GetNumberOfItems() == 1 );
Item &item2 = sqi2->GetItem(1);
DataSet &ds3 = item2.GetNestedDataSet();

const PrivateTag tzlib(0x200d,0x3cfa,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tzlib ) ) return 1;
const DataElement& zlib = ds3.GetDataElement( tzlib );

const ByteValue *bv = zlib.GetByteValue();
if( !bv ) return 1;
if( bv->GetLength() != 4 ) return 1;

// (200d,3010) IS 2 88
const PrivateTag tnslices(0x200d,0x3010,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tnslices ) ) return 1;
const DataElement& nslices = ds3.GetDataElement( tnslices );
Element<VR::IS,VM::VM1> elnslices;
elnslices.SetFromDataElement( nslices );
const int nslicesref = elnslices.GetValue();
gdcmm_assert( nslicesref >= 0 );
// (200d,3011) IS 6 259648
const PrivateTag tzalloc(0x200d,0x3011,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzalloc ) ) return 1;
const DataElement& zalloc = ds3.GetDataElement( tzalloc );
Element<VR::IS,VM::VM1> elzalloc;
elzalloc.SetFromDataElement( zalloc );
const int zallocref = elzalloc.GetValue();
gdcmm_assert( zallocref >= 0 );
// (200d,3021) IS 2 0
const PrivateTag tzero(0x200d,0x3021,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzero ) ) return 1;
const DataElement& zero = ds3.GetDataElement( tzero );
Element<VR::IS,VM::VM1> elzero;
elzero.SetFromDataElement( zero );
const int zeroref = elzero.GetValue();
gdcmm_assert( zeroref == 0 ); (void)zeroref;

// (200d,3cf3) OB
const PrivateTag tdeflate(0x200d,0x3cf3,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tdeflate ) ) return 1;
const DataElement& deflate = ds3.GetDataElement( tdeflate );
const ByteValue *bv2 = deflate.GetByteValue();

// (200d,3cfb) OB
const PrivateTag tcrc(0x200d,0x3cfb,"Philips US Imaging DD 045");
if( !ds3.FindDataElement( tcrc ) ) return 1;
const DataElement& crc = ds3.GetDataElement( tcrc );
const ByteValue *bv3 = crc.GetByteValue();

std::string outfile = std::string( bvdatatype->GetPointer(), bvdatatype->GetLength() );
outfile = LOComp::Trim( outfile.c_str() );
const char *outfilename = outfile.c_str();
#ifdef NDEBUB
gdcmm_assert( is_valid(outfilename) );
#endif
if( bv2 )
{
gdcmm_assert( bv3 );
gdcmm_assert( zallocref > 0 );
gdcmm_assert( nslicesref > 0 );
std::cout << ds2 << std::endl;

if( strcmp(bv->GetPointer(), "ZLib", 4) == 0 )
{
if( !ProcessDeflate( outfile, nslicesref, zallocref, bv2->GetPointer(),
std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
{
return 1;
}
}
}

```

```

    }
}
else if( strcmp(bv->GetPointer(), "None", 4) == 0 )
{
    if( !ProcessNone( outfilename, nslicesref, zallocref, bv2->GetPointer(),
        std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
    {
        return 1;
    }
}
else
{
    std::string str( bv->GetPointer(), bv->GetLength() );
    std::cerr << "Unhandled: " << str << std::endl;
    return 1;
}
}
}

return 0;
}

```

14.55 DumpSiemensBase64.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ2ILP8EcM/WzjFrtjnAgAJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include "gdcmCSAHeader.h"
#include "gdcmBase64.h"
#include "gdcmExplicitDataElement.h"
#include "gdcmSwapper.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    gdcm::CSAHeader csa;
    const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    if( !ds.FindDataElement( t1 ) ) return 1;
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );

    //const char name[] = "MRDiffusion";
    const char name[] = "MR_AS_L";
    if( csa.FindCSAElementByName(name) )
    {

```

```

const gdcM::CSAElement & el = csa.GetCSAElementByName(name);
const gdcM::ByteValue* bv = el.GetByteValue();
std::string str( bv->GetPointer(), bv->GetLength() );
str.erase(std::remove(str.begin(), str.end(), '\n'), str.end());
size_t dl = gdcM::Base64::GetDecodeLength( str.c_str(), str.size() );
std::vector<char> buf;
buf.resize( dl );
size_t dl2 = gdcM::Base64::Decode( buf.data(), buf.size(), str.c_str(), str.size() );
(void)dl2;
std::stringstream ss;
ss.str( std::string(buf.data(), buf.size()) );
gdcM::File file;
gdcM::DataSet &ds2 = file.GetDataSet();
gdcM::DataElement xde;
try
{
    while( xde.Read<gdcM::ExplicitDataElement,gdcM::SwapperNoOp>( ss ) )
    {
        ds2.Insert( xde );
    }
    gdcM__assert( ss.eof() );
}
catch(std::exception &e)
{
    return 1;
}
gdcM::Printer p;
p.SetFile( file );
p.Print(std::cout);
}

return 0;
}

```

14.56 DumpToSQLITE3.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Ref:
* http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/004408.html
* Implementation details:
* http://www.sqlite.org/c3ref/bind_blob.html
* http://www.adp-gmbh.ch/sqlite/bind_insert.html
*/
#include "gdcMScanner.h"
#include "gdcMDirectory.h"
#include "gdcMTag.h"
#include "gdcMTrace.h"

#include "sqlite3.h"

#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    time_t time_start = time(nullptr);

    gdcM::Trace::SetDebug( false );

```

```

gdcmm::Trace::SetWarning( false );
const char *inputdirectory = argv[1];

gdcmm::Directory d;
unsigned int nfiles = d.Load( inputdirectory, true);

gdcmm::Scanner s;
using gdcmm::Tag;
s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

bool b0 = s.Scan( d.GetFilenames() );
if( !b0 ) return 1;
time_t time_scanner = time(nullptr);

std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;

// MappingType const &mappings = s.GetMappings();

sqlite3* db;
sqlite3_open("./dicom.db", &db);

if(db == nullptr)
{
    std::cerr << "Could not open database." << std::endl;
    return 1;
}

const char sql_stmt[] = "create table browser (seriesuid, studyuid)";
int ret;

char *errmsg;
ret = sqlite3_exec(db, sql_stmt, nullptr, nullptr, &errmsg);

if(ret != SQLITE_OK)
{
    printf("Error in statement: %s [%s].\n", sql_stmt, errmsg);
    return 1;
}
using gdcmm::Directory;
using gdcmm::Scanner;
const Directory::FilenamesType& files = d.GetFilenames();
Directory::FilenamesType::const_iterator file = files.begin();

sqlite3_stmt *stmt;
if ( sqlite3_prepare(
    db,
    "insert into browser values (?,?)", // stmt
    -1, // If than zero, then stmt is read up to the first nul terminator
    &stmt,
    nullptr // Pointer to unused portion of stmt
)
!= SQLITE_OK)
{
    printf("\nCould not prepare statement.");
    return 1;
}
//printf("\nThe statement has %d wildcards\n", sqlite3_bind_parameter_count(stmt));
for(; file != files.end(); ++file)
{
    const char *filename = file->c_str();
    bool b = s.IsKey(filename);
    if( b )
    {
        const Scanner::TagToValue &mapping = s.GetMapping(filename);
        Scanner::TagToValue::const_iterator it = mapping.begin();

        sqlite3_reset(stmt);

        for( int index = 1; it != mapping.end(); ++it, ++index)
        {
            //const Tag & tag = it->first;
            const char *value = it->second;

            if (sqlite3_bind_text (
                stmt,
                index, // Index of wildcard
                value,
                (int)strlen(value), // length of text
                SQLITE_STATIC // SQLite assumes that the information is in static
            )

```

```

    )
    != SQLITE_OK)
    {
        printf("\nCould not bind int.\n");
        return 1;
    }
}
if (sqlite3_step(stmt) != SQLITE_DONE)
{
    printf("\nCould not step (execute) stmt.\n");
    return 1;
}
}
}

sqlite3_close(db);

time_t time_sqlite = time(nullptr);

std::cout << "Time to scan DICOM files: " << (time_scanner - time_start) << std::endl;
std::cout << "Time to build SQLITE3: " << (time_sqlite - time_scanner) << std::endl;

return 0;
}

```

14.57 DumpToshibaDTI.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu81VvAMJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI( const char * input, size_t len )
{
    static int i = 0;
    ++i;
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    #if 0
    std::ostream f;
    f << "debug" << i;
    std::ofstream of( f.str().c_str(), std::ios::binary );
    of.write( &copy[0], copy.size() );
    of.close();
    #else

    std::stringstream is;
    std::string dup( copy.data(), copy.size() );
    is.str( dup );

    gdcm::File file;
    gdcm::FileMetaInformation & fmi = file.GetHeader();

```

```

fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );
gdcmm::DataSet & ds = file.GetDataSet();
ds.Read<gdcmm::ExplicitDataElement,gdcmm::SwapperNoOp>( is );

//gdcmm::DictPrinter p;
gdcmm::Printer p;
p.SetFile( file );
p.SetColor( true );
p.Print( std::cout );
#endif

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

    // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]           # 22,1 Private Creator
    // (0029,1001) ?? (SQ) (Sequence with undefined length)    # u/1,1 ?

    const gdcmm::PrivateTag tpmf(0x0029,0x1,"PMTF INFORMATION DATA");
    if( !ds.FindDataElement( tpmf ) ) return 1;
    const gdcmm::DataElement& pmtf = ds.GetDataElement( tpmf );
    if ( pmtf.IsEmpty() ) return 1;
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> seq = pmtf.GetValueAsSQ();
    if ( !seq || !seq->GetNumberOfItems() ) return 1;

    size_t n = seq->GetNumberOfItems();
    for( size_t i = 1; i <= n; ++i )
    {
        gdcmm::Item &item = seq->GetItem(i);
        gdcmm::DataSet &subds = item.GetNestedDataSet();
        // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]           # 22,1 Private Creator
        // (0029,1090) ?? (OB) 00\05\00\13\00\12\00\22\           # 202,1 ?
        const gdcmm::PrivateTag tseq(0x0029,0x90,"PMTF INFORMATION DATA");

        if( subds.FindDataElement( tseq ) )
        {
            const gdcmm::DataElement &de = subds.GetDataElement( tseq );
            const gdcmm::ByteValue *bv = de.GetByteValue();
            if( !bv ) return 1;

            bool b = DumpToshibaDTI( bv->GetPointer(), bv->GetLength() );
            if( !b ) return 1;
        }
    }

    return 0;
}

```

14.58 DumpToshibaDTI2.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

```

```

* https://gazelle.ihe.net/EVSCClient/dicomResult.seam;jses-
  sionid=x+Rf9Zs+ip49P+jC3L8SLZb8?&oid=1.3.6.1.4.1.12559.11.1.2.1.4.1622284
*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI2( const char * input, size_t len )
{
    static int i = 0;
    ++i;
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    #if 0
        std::ostream f;
        f << "debug" << i;
        std::ofstream of( f.str().c_str(), std::ios::binary );
        of.write( &copy[0], copy.size() );
        of.close();
    #else

        std::stringstream is;
        std::string dup( copy.data(), copy.size() );
        is.str( dup );

        gdcm::File file;
        gdcm::FileMetaInformation & fmi = file.GetHeader();
        fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
        gdcm::DataSet & ds = file.GetDataSet();
        ds.Read<gdcm::ExplicitDataElement, gdcm::SwapperNoOp>( is );

        //gdcm::DictPrinter p;
        gdcm::Printer p;
        p.SetFile( file );
        p.SetColor( true );
        p.Print( std::cout );
    #endif

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    /*
(0029,1001) SQ (Sequence with explicit length ==6)      # 18746, 1 Unknown Tag & Data
(fffe,e000) na (Item with explicit length ==2)          # 206, 1 Item
  (0029,0010) LO [TOSHIBA_MEC_MR3]                      # 16, 1 PrivateCreator
  (0029,1090) OB 00\07\00\06\00\05\00\04\00\03\00\02\00\0c\00\01\00\00\00\00\12... # 170, 1 Unknown Tag & Data
(fffe,e00d) na (ItemDelimitationItem for re-encoding)  # 0, 0 ItemDelimitationItem
(fffe,e000) na (Item with explicit length ==2)          # 866, 1 Item
  (0029,0010) LO [TOSHIBA_MEC_MR3]                      # 16, 1 PrivateCreator
  (0029,1090) OB 45\4e\49\50\53\4c\20\52\41\5c\45\4e\49\50\53\4c\54\5c\52\45\53\55... # 830, 1 Unknown Tag & Data
[...]
(0029,1002) SQ (Sequence with explicit length ==1)      # 120, 1 Unknown Tag & Data
(fffe,e000) na (Item with explicit length ==2)          # 112, 1 Item
  (0029,0010) LO [TOSHIBA_MEC_MR3]                      # 16, 1 PrivateCreator
  (0029,1090) OB 00\10\00\02\53\55\10\80\70\0d\30\31\5e\33\52\4d\5f\43\45\4d\5f\41... # 76, 1 Unknown Tag & Data
(fffe,e00d) na (ItemDelimitationItem for re-encoding)  # 0, 0 ItemDelimitationItem

```

```

*/

const gdcm::PrivateTag tmeclr3(0x0029,0x1,"TOSHIBA_MEC_MR3");
if( !ds.FindDataElement( tmeclr3 ) ) return 1;
const gdcm::DataElement& meclr3 = ds.GetDataElement( tmeclr3 );
if ( meclr3.IsEmpty() ) return 1;
gdcm::SmartPointer<gdcm::SequenceOfItems> seq = meclr3.GetValueAsSQ();
if ( !seq || !seq->GetNumberOfItems() ) return 1;

size_t n = seq->GetNumberOfItems();
for( size_t i = 1; i <= n; ++i )
{
    gdcm::Item &item = seq->GetItem(i);
    gdcm::DataSet &subds = item.GetNestedDataSet();
    const gdcm::PrivateTag tseq(0x0029,0x90,"TOSHIBA_MEC_MR3");

    if( subds.FindDataElement( tseq ) )
    {
        const gdcm::DataElement &de = subds.GetDataElement( tseq );
        const gdcm::ByteValue *bv = de.GetByteValue();
        if( !bv ) return 1;

        bool b = DumpToshibaDTI2( bv->GetPointer(), bv->GetLength() );
        if( !b ) return 1;
    }

}

return 0;
}

```

14.59 DumpVisusChange.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmDirectory.h"
#include "gdcmStringFilter.h"

#include <vector>
#include <algorithm>

/*
*/
static bool process( std::vector<gdcm::DataElement> & ms, const char * filename)
{
    using namespace gdcm;
    Tag pd(0x7fe0,0x0000);
    std::set<gdcm::Tag> skiptags;
    skiptags.insert( pd );

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.ReadUpToTag( pd, skiptags ) )
    {
        std::cerr << "Failure to read: " << filename << std::endl;
        return false;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds1 = file.GetDataSet();

    const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
    if( !ds1.FindDataElement( tseq1 ) ) return true;
}

```



```

const gdcmm::DataElement& seq1 = ds1.GetDataElement( tseq1 );

SmartPointer<SequenceOfItems> sqi1 = seq1.GetValueAsSQ();

const size_t nitems = sqi1->GetNumberOfItems();
for( size_t item = 1; item < nitems; ++item )
{
    Item &item1 = sqi1->GetItem(item);
    DataSet &ds2 = item1.GetNestedDataSet();
    for(DataSet::ConstIterator it = ds2.Begin(); it != ds2.End(); ++it )
    {
        DataElement const & de = *it;
        // cannot simply use std::set here, see there is a discrepancy in between
        // operator== and operator<.
        // So only use operator== here:
        std::vector<DataElement>::iterator vit = std::find(ms.begin(), ms.end(), de);
        if( vit == ms.end() )
            ms.push_back(de);
    }
}
return true;
}

int main(int argc, char *argv[])
{
    bool usefastpath = true;

    if( argc < 2 ) return 1;
    using namespace gdcmm;
    const char *filename = argv[1];
    gdcmm::Directory::FileNamesType filenames;
    if( !gdcmm::System::FileExists(filename) )
    {
        std::cerr << "Could not find file: " << filename << std::endl;
        return 1;
    }

    gdcmm::Directory dir;
    if( gdcmm::System::FileIsDirectory(filename) )
    {
        unsigned int nfiles = dir.Load(filename, false);
        if( nfiles == 0 )
        {
            std::cerr << "Could not find files: " << filename << std::endl;
            return 1;
        }
        filenames = dir.GetFileNames();
    }
    else
    {
        filenames.push_back( filename );
    }
    gdcmm::StringFilter sf;

    Tag pd(0x7fe0,0x0000);
    std::set<gdcmm::Tag> skiptags;
    skiptags.insert( pd );

    gdcmm::Reader reader;
    reader.SetFileName( filenames[0].c_str() );
    if( !reader.ReadUpToTag( pd, skiptags ) )
    {
        std::cerr << "Could not read file: " << filename << std::endl;
        return 1;
    }
    gdcmm::File &file = reader.GetFile();
    sf.SetFile(file);

    if( usefastpath ) {
        // Heuristic, assume if private tag cannot be found in first file, skip the directory
        gdcmm::DataSet &ds1 = file.GetDataSet();

        const gdcmm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
        if( !ds1.FindDataElement( tseq1 ) ){
            std::cerr << "Could not find private tag in first file skipping whole directory: " << filename << std::endl;
            return 0;
        }
    }
    std::vector<DataElement> ms;
    for(gdcmm::Directory::FileNamesType::const_iterator cit = filenames.begin(); cit != filenames.end(); ++cit )
    {

```

```

    if( !process(ms, cit->c_str()) ) {
        return 1;
    }
}

if( !ms.empty() ) {
    std::sort(ms.begin(), ms.end());
    std::cout << filename << ",\n";
    for(std::vector<DataElement>::const_iterator it = ms.begin(); it != ms.end(); ++it )
    {
        DataElement const & de = *it;
        std::string const & s = sf.ToString( de );
        std::cout << de.GetTag() << " " << s << std::endl;
    }
    std::cout << "\n" << std::endl;
}

return 0;
}

```

14.60 DuplicatePCDE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
Usage:
DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:
http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af

Question:
1.
    There is no restriction for a specific Private Creator Data Element
    (PCDE) to be unique within the same group, right ?
    Decoders of Private Data would have to handle the case where a PCDE
    would be repeated and should NOT stop on the first instance of a
    particular PCDE, right ?

    Eg. when searching for the tag associated with
    (0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
    dataset:

(0029,0010) LO [SIEMENS CSA HEADER]          # 18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER]        # 22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2]       # 22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER]          # 18, 1
PrivateCreator

```

```

(0029,1008) CS [IMAGE NUM 4] # 12, 1
CSAImageHeaderType
(0029,1009) LO [20050723] # 8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR] # 2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723] # 8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300] # 14, 1
PMTFInformation1
(0029,1132) UL 32768 # 4, 1
PMTFInformation2
(0029,1133) UL 0 # 4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM] # 12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20 # 4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo

```

one should return two instances, correct ?

Answer:

I would say that this is covered in principle by the PS 3.5 7.1
 "The Data Elements ... shall occur at most once in a Data Set"
 rule, since the data element is defined by the tuple
 (private creator,gggg,ee) where xxee is the element
 number and xx is arbitrary and has no inherent meaning and
 does not serve to disambiguate the data element.

E.g.:

```

(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"

```

would be illegal because even though they are assigned different
 (completely arbitrary) blocks, with the same group, element
 number and private creator, (0019,3015) and (0019,3215) are the
 "same" data element.

*/

```

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Let's get all private element from group 0x9:
    /*
(0009,0010) LO [GEMS_IDEN_01] # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ] # 14,1 Full fidelity
(0009,1002) SH [CT01] # 4,1 Suite id
(0009,1004) SH [HiSpeed CT/i] # 12,1 Product id
(0009,1027) SL 862399669 # 4,1 Image actual date
(0009,1030) SH (no value) # 0,1 Service id
(0009,1031) SH (no value) # 0,1 Mobile location number
(0009,10e6) SH [05] # 2,1 Genesis Version - now

```

```

(0009,10e7) UL 973283917                                # 4,1 Exam Record checksum
(0009,10e9) SL 862399669                                # 4,1 Actual series data time stamp
*/
gdcmm::Tag start(0x0009,0x0);
// Create a temporary duplicate dataset, since we cannot insert data element as we go over them (std::set
// would reorganize itself as we go over it ...)
gdcmm::DataSet dup;
gdcmm::Tag new_private(0x0009,0x0);
while (start.GetGroup() == 0x9 )
{
    const gdcmm::DataElement& de = ds.FindNextDataElement(start);
    const gdcmm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
    {
        std::cout << t << std::endl;
        // Ok let's duplicate into the next available attribute:
        gdcmm::DataElement duplicate = de;
        duplicate.GetTag().SetElement( (uint16_t)(t.GetElement() + 1) );
        dup.Insert( duplicate );
        new_private = duplicate.GetTag();
    }
    else if( t.IsPrivate() && !t.IsPrivateCreator() )
    {
        //std::cout << de << std::endl;
        std::string owner = ds.GetPrivateCreator( de.GetTag() );
        //std::cout << owner << std::endl;
        gdcmm::DataElement duplicate = de;
        duplicate.GetTag().SetPrivateCreator( new_private );
        if( const gdcmm::ByteValue *bv = duplicate.GetByteValue() )
        {
            // Warning: when doing : duplicate = de, only the pointer to the ByteValue is passed
            // (to avoid large memory duplicate). We need to explicitly duplicate the bytevalue ourselves:
            gdcmm::ByteValue *dupbv = new gdcmm::ByteValue( bv->GetPointer(),
                bv->GetLength() );
            // Let's recognize the duplicated ASCII-type elements:
            if( duplicate.GetVR() & gdcmm::VR::VRASCII )
                dupbv->Fill( 'X' );
            duplicate.SetValue( *dupbv );
        }
        dup.Insert( duplicate );
    }
    start = t;
    // move to next possible 'public' element
    start.SetElement( (uint16_t)(start.GetElement() + 1) );
}

gdcmm::DataSet::ConstIterator it = dup.Begin();
for( ; it != dup.End(); ++it )
{
    ds.Insert( *it );
}

gdcmm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

14.61 ELSCINT1WaveToText.cxx

```
/*=====
```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcmm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

===== */
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"

/*
 * This example shows how to read a Wave Information tag from ELSCINT1
 * The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
 * Secondary Capture Image Storage (usually a 'N' Symbol is shown)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcml-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Gauthier Bouilhol
 */

template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8)
{
    static const char sep = '\t';
    os << c1 << sep << c2 << sep << c3 << sep << c4 << sep << c5 << sep << c6 << sep << c7 << sep << c8;
    os << std::endl;
    return true;
}

bool wave2stream( std::ostream &text_file, const char *in, size_t len )
{
    const short * buffer = (const short*)in;
    size_t length = len / sizeof( short );
    text_file << "COMPLETE_WAVE" << '\t' << "MASK" << '\t' << "AQUISITION_PROFIL" << '\t' << "END-INHALE" << '\t' <<
        "END-EXHALE" << '\t' << "AQUISITION_WAVE" << '\t' << "WAVE_STATISTICS" << '\t' << "MASK" << std::endl;
    for (size_t i=0;i<length-76;i+=2)
    {
        if ( i < 74 )
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == 256)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == -32768)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == -16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == -32512)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << buffer[i] << '\t' << buffer[i+1] << std::endl;
        }
        else
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == 256)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == -32768)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == -16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == -32512)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
        }
    }
}

```

```

    }

    return true;
}

int main(int argc, char *argv [])
{
    if( argc < 3 ) return 1;
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
    if( !ds.FindDataElement( twave ) ) return 1;
    const gdcm::DataElement& wave = ds.GetDataElement( twave );
    if ( wave.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = wave.GetByteValue();
    gdcm_assert( bv );

    std::ofstream os( outfile, std::ios::binary );
    // Dump that to a CSV file:
    wave2stream( os, bv->GetPointer(), bv->GetLength() );
    os.close();

    return 0;
}

```

14.62 EmptyMask.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmEmptyMaskGenerator.h"

#include <string>
#include <cstring>

int main( int argc, char *argv[] )
{
    std::string inputdir;
    std::string outputdir;
    bool input_sopclassuid = true;
    bool grayscale_secondary_sopclassuid = false;
    if( argc < 3 ) return 1;
    inputdir = argv[1];
    outputdir = argv[2];
    // input_sopclassuid -> Use original SOP Class UID from input DICOM (Default).
    // grayscale_secondary_sopclassuid -> Use Grayscale Secondary Image Storage SOP Class UID.
    if( argc >= 3 )
    {
        input_sopclassuid = false;
        if( strcmp("input_sopclassuid", argv[3]) == 0 )
            input_sopclassuid = true;
        else if( strcmp("grayscale_secondary_sopclassuid", argv[3]) == 0 ) {
            grayscale_secondary_sopclassuid = true;
        }
    }

    //
    gdcm::EmptyMaskGenerator emg;

```

```

if( input_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseOriginalSOPClassUID );
else if( grayscale_secondary_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseGrayscaleSecondaryImageStorage );
emg.SetInputDirectory( inputdir.c_str() );
emg.SetOutputDirectory( outputdir.c_str() );
if( !emg.Execute() )
{
    return 1;
}

return 0;
}

```

14.63 EncapsulateFileInRawData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"

#include "magic.h" // libmagic, API to file command line tool

/*
 * Let say you want to encapsulate a file type that is not defined in DICOM (exe, zip, png)
 * PNG is a bad example, unless it contains transparency (which has been deprecated).
 * It will take care of dispatching each chunk to an appropriate data item (pretty much like
 * WaveformData)
 *
 * Usage:
 * ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
 */

// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " inputfile output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    if( !gdcm::System::FileExists( filename ) ) return 1;

    size_t s = gdcm::System::FileSize(filename);
    if( !s ) return 1;

    magic_t cookie = magic_open(MAGIC_NONE);
    const char * file_type = magic_file(cookie, filename);
    if( !file_type ) return 1;
    magic_close(cookie);

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    //gdcm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );

```

```

w.SetFileName( outfilename );

file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );

gdcm::Anonymizer anon;
anon.SetFile( file );

gdcm::MediaStorage ms = gdcm::MediaStorage::RawDataStorage;

gdcm::UIDGenerator gen;
anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
std::cout << ms.GetString() << std::endl;
anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );

if (!w.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

14.64 ExtractEncryptedContent.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"

#include <fstream>

/*

openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out outputfile.der -outform DER
../trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm -outform DER -inkey
../trunk/Testing/Source/Data/privatekey.pem ../trunk/Testing/Source/Data/certificate.pem

*/

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.der" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();

```



```

gdcmm::DataSet &ds = file.GetDataSet();

const gdcmm::DataElement &EncryptedAttributesSequence = ds.GetDataElement( gdcmm::Tag( 0x0400,0x0500 ) );

gdcmm::SequenceOfItems *sqi = EncryptedAttributesSequence.GetValueAsSQ();

if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

gdcmm::Item &item = sqi->GetItem(1);

gdcmm::DataSet &nesteddds = item.GetNestedDataSet();

if( ! nesteddds.FindDataElement( gdcmm::Tag( 0x0400,0x0520 ) ) ) return 1;

const gdcmm::DataElement &EncryptedContent = nesteddds.GetDataElement( gdcmm::Tag( 0x0400,0x0520 ) );

const gdcmm::ByteValue *bv = EncryptedContent.GetByteValue();

std::ofstream of( outfilename, std::ios::binary );
of.write( bv->GetPointer(), bv->GetLength() );
of.close();

return 0;
}

```

14.65 ExtractIconFromFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmmImageReader.h"
#include "gdcmmPNMCodec.h"
#include "gdcmmIconImageFilter.h"
#include "gdcmmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcmm::IconImage& icon)
{
    gdcmm::PNMCodec pnm;
    pnm.SetDimensions( icon.GetDimensions() );
    pnm.SetPixelFormat( icon.GetPixelFormat() );
    pnm.SetPhotometricInterpretation( icon.GetPhotometricInterpretation() );
    pnm.SetLUT( icon.GetLUT() );
    const gdcmm::DataElement& in = icon.GetDataElement();
    bool b = pnm.Write( filename, in );
    gdcmm::assert(b);
    return b;
}

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcmm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read (or not image): " << filename << std::endl;
        return 1;
    }

    gdcmm::IconImageFilter iif;
    iif.SetFile( reader.GetFile() );

```

```

bool b = iif.Extract();

if( b )
{
    const gdcm::IconImage &icon = iif.GetIconImage(0);
    icon.Print( std::cout );

    if( !icon.GetTransferSyntax().IsEncapsulated() )
    {
        // Let's write out this icon as PNM file
        WriteIconAsPNM("icon.ppm", icon);
    }
    else if( icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGBaselineProcess1
    || icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGExtendedProcess2_4
    )
    {
        const gdcm::DataElement& in = icon.GetDataElement();
        const gdcm::ByteValue *bv = in.GetByteValue();
        gdcm_assert( bv );
        std::ofstream out( "icon.jpg", std::ios::binary );
        out.write( bv->GetPointer(), bv->GetLength() );
        out.close();
    }
}
else
{
    gdcm_assert( iif.GetNumberOfIconImages() == 0 );
    std::cerr << "No Icon Found anywhere in file" << std::endl;

    const gdcm::Image &img = reader.GetImage();
    gdcm::IconImageGenerator iig;
    iig.AutoPixelMinMax(true);
    iig.SetPixmap( img );
    const unsigned int idims[2] = { 64, 64 };
    iig.SetOutputDimensions( idims );
    //iig.SetPixelMinMax(60, 868);
    if( !iig.Generate() ) return 1;
    const gdcm::IconImage & icon = iig.GetIconImage();
    WriteIconAsPNM("icon.ppm", icon);
}

return 0;
}

```

14.66 Extracting_All_Resolution.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include <fstream>
#include <stdint.h>
#include <string.h>
#include <assert.h>
#include <gdcm_j2k.h>
#include <gdcm_jp2.h>
#include <iostream>
#include <cstring>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmSystem.h"

```

```

#include <fstream>

#include "gdcm_openjpeg.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

void error_callback(const char *msg, void *) {
    (void)msg;
}
void warning_callback(const char *msg, void *) {
    (void)msg;
}
void info_callback(const char *msg, void *) {
    (void)msg;
}

bool Write_Resolution(gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res, std::ostream& of, int flag,
    gdcm::SequenceOfItems *sq, int No_Of_Resolutions)
{
    std::ifstream is;
    is.open( filename, std::ios::binary );
    opj_dparameters_t parameters; /* decompression parameters */
    opj_event_mgr_t event_mgr; /* event manager */
    opj_dinfo_t* dinfo; /* handle to a decompressor */
    opj_cio_t *cio;
    opj_image_t *image = NULL;
    // FIXME: Do some stupid work:
    is.seekg( 0, std::ios::end);
    std::streampos buf_size = is.tellg();
    char *dummy_buffer = new char[(unsigned int)buf_size];
    is.seekg(0, std::ios::beg);
    is.read( dummy_buffer, buf_size);
    unsigned char *src = (unsigned char*)dummy_buffer;
    uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok since DICOM cannot have larger than 2Gb image

    /* configure the event callbacks (not required) */
    memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
    event_mgr.error_handler = error_callback;
    event_mgr.warning_handler = warning_callback;
    event_mgr.info_handler = info_callback;

    /* set decoding parameters to default values */
    opj_set_default_decoder_parameters(&parameters);

    // default blindly copied
    parameters.cp_layer=0;
    parameters.cp_reduce= res;
    // parameters.decod_format=-1;
    // parameters.cod_format=-1;

    const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
    if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
    {
        /* JPEG-2000 compressed image data ... sigh */
        // gdcmData/ELSCINT1_JP2vsJ2K.dcm
        // gdcmData/MAROTECH_CT_JP2Lossy.dcm
        //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
        parameters.decod_format = 1; //JP2_CFMT;
        //gdcm_assert(parameters.decod_format == JP2_CFMT);
    }
    else
    {
        /* JPEG-2000 codestream */
        //parameters.decod_format = J2K_CFMT;
        //gdcm_assert(parameters.decod_format == J2K_CFMT);
        gdcm_assert( 0 );
    }
}

```

```

    }
    parameters.cod_format = 11; // PGX_DFMT;
    //gdcm_assert(parameters.cod_format == PGX_DFMT);

    /* get a decoder handle */
    dinfo = opj_create_decompress(CODEC_JP2);

    /* catch events using our callbacks and give a local context */
    opj_set_event_mgr((opj_common_ptr)dinfo, &event_mgr, NULL);

    /* setup the decoder decoding parameters using user parameters */
    opj_setup_decoder(dinfo, &parameters);

    /* open a byte stream */
    cio = opj_cio_open((opj_common_ptr)dinfo, src, file_length);

    /* decode the stream and fill the image structure */
    image = opj_decode(dinfo, cio);
    if(!image) {
        opj_destroy_decompress(dinfo);
        opj_cio_close(cio);
        //gdcmErrorMacro( "opj_decode failed" );
        return 1;
    }

    opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
    opj_tcp_t * tcp = &cp->tcps[0];
    opj_tccp_t * tccp = &tcp->tccps[0];
    /* std::cout << "\n No of Cols In Image" << image->x1;
    std::cout << "\n No of Rows In Image" << image->y1;
    std::cout << "\n No of Components in Image" << image->numcomps;
    std::cout << "\n No of Resolutions" << tccp->numresolutions << "\n";
    */
    opj_j2k_t * j2k = NULL;
    opj_jp2_t * jp2 = NULL;
    jp2 = (opj_jp2_t*)dinfo->jp2_handle;
    int reversible = jp2->j2k->cp->tcps->tccps->qmfbid;
    //std::cout << reversible;
    int compno = 0;
    opj_image_comp_t *comp = &image->comps[compno];
    int Dimensions[2];
    Dimensions[0] = comp->w;
    Dimensions[1] = comp->h;
    opj_cio_close(cio);
    unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
    //std::cout << "\nTest" << image->comps[0].factor;
    char *raw = new char[len];
    for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno++)
    {
        opj_image_comp_t *comp = &image->comps[compno];

        int w = image->comps[compno].w;
        int h = image->comps[compno].h;
        uint8_t *data8 = (uint8_t*)raw + compno;
        for (int i = 0; i < w * h ; i++)
        {
            int v = image->comps[compno].data[i];
            *data8 = (uint8_t)v;
            data8 += image->numcomps;
        }
    }
}

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
de1.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage );
de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));

```

```

ds.Insert( de1 );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0010> row = {image->comps[0].w};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
ds.Insert( col.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {image->numcomps};
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

if (flag == 1)
{
    for (int i=0; i < No_Of_Resolutions; i++)
    {
        int a = 1;
        int b = 1;

        while(a!==(No_Of_Resolutions)-i)
        {
            b = b*2;
            a = a+1;
        }
        uint16_t row = (image->y1)/b;
        uint16_t col = (image->x1)/b;
        //std::cout << row;
        gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1> el2;
        el2.SetValue(i+1);
        gdcm::DataElement rfn = el2.GetAsDataElement();    //ulr --> upper left row
        rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

        gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el;
        el.SetValue(1,0);
        el.SetValue(1,1);
        gdcm::DataElement ulr = el.GetAsDataElement();    //ulr --> upper left col/row
        ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

        gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el1;
        el1.SetValue(col,0);
        el1.SetValue(row,1);
        gdcm::DataElement brr = el1.GetAsDataElement();
        brr.SetTag( gdcm::Tag(0x0048,0x0202) );    //brr --> bottom right col/row
        gdcm::Item it;
        gdcm::DataSet &nds = it.GetNestedDataSet();
        nds.Insert( rfn );
        nds.Insert( ulr );
        nds.Insert( brr );

        sq->AddItem(it);
    }

    gdcm::Writer w1;
    gdcm::File &file1 = w1.GetFile();
    gdcm::DataSet &ds1 = file1.GetDataSet();
    file1.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::UIDGenerator uid1;
    gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); // SOP Instance UID

```

```

    dea.SetVR( gdcmm::VR::UI );
    const char *u1 = uid1.Generate();
    dea.SetByteValue( u1, strlen(u1) );
    ds1.Insert( dea );

    gdcmm::DataElement deb( gdcmm::Tag(0x8,0x16) );
    deb.SetVR( gdcmm::VR::UI );
    gdcmm::MediaStorage ms1( gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
    deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()));
    ds1.Insert( deb );

    const char mystr1[] = "MONOCHROME2 ";
    gdcmm::DataElement dec( gdcmm::Tag(0x28,0x04) );
    //de.SetTag(gdcmm::Tag(0x28,0x04));
    dec.SetVR( gdcmm::VR::CS );
    dec.SetByteValue(mystr, strlen(mystr));
    ds1.Insert( dec );

    gdcmm::Attribute<0x0028,0x0010> row1 = {image->y1};
    //row.SetValue(512);
    ds1.Insert( row1.GetAsDataElement() );
    // w.SetCheckFileMetaInformation( true );
    gdcmm::Attribute<0x0028,0x0011> col1 = {image->x1};
    ds1.Insert( col1.GetAsDataElement() );
    gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
    ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

    gdcmm::Attribute<0x0028,0x0100> ata = {8};
    ds1.Insert( ata.GetAsDataElement() );

    gdcmm::Attribute<0x0028,0x0002> atb = {image->numcomps};
    ds1.Insert( atb.GetAsDataElement() );

    gdcmm::Attribute<0x0028,0x0101> atc = {8};
    ds1.Insert( atc.GetAsDataElement() );

    gdcmm::Attribute<0x0028,0x0102> atd = {7};
    ds1.Insert( atd.GetAsDataElement() );

    theStreamWriter.SetFile(file1);

    gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
    des.SetVR(gdcmm::VR::SQ);
    //des.SetVR(gdcmm::VM::VM1);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds1.Insert(des);

    if (!theStreamWriter.WriteImageInformation()){
        std::cerr << "unable to write image information" << std::endl;
        return 1; //the CanWrite function should prevent getting here, else,
        //that's a test failure
    }
}

theStreamWriter.SetFile(file);

if (!theStreamWriter.CanWriteFile()){
    delete [] raw;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout << "\nabletoread";

// Important to write here
std::vector<unsigned int> extent = gdcmm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 4;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];
    std::cout << "\n" << xmax << "\n" << ymax << "\n" << zmax << "\n" << image->numcomps << "\n";

    if (xmax == 0 || ymax == 0)

```

```

    {
        std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
        return 0;
    }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially. So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
        for (y = 0; y < ymax; y += ychunk){
            nexty = y + ychunk;
            if (nexty > ymax) nexty = ymax;
            theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
            unsigned long len = theStreamWriter.DefineProperBufferLength();
            std::cout << "\n" << len;
            char* finalBuffer = new char[len];
            memcpy(finalBuffer, &(raw[prevLen]), len);
            std::cout << "\nable to write";
            if (!theStreamWriter.Write(finalBuffer, len)){
                std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z << std::endl;
                delete [] raw;
                delete [] finalBuffer;
                return 1;
            }
            delete [] finalBuffer;
            prevLen += len;
        }
    }
    delete raw;

    delete[] src; //FIXME

    if(dinfo) {
        opj__destroy__decompress(dinfo);
    }

    opj_image__destroy(image);

    return true;
}

bool Different_Resolution( gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res, std::ostream& of)
{
    //std::vector<std::string>::const_iterator it = filenames.begin();
    bool b = true;
    int flag = 1;

    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    for(int i = res-1 ; i>=0; --i)
    {
        b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
        // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
        flag = 0;
    }
    //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
    //b = b && PopulateSingeFile( writer, sq, jpeg, filename2 );
    //image.SetDimension(2, res )
    return b;
}

int main(int argc, char *argv[])
{
    if( argc < 4 )
    {
        std::cerr << argv[0] << " input.jp2 output.dcm No. Of Resolutions " << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    char *resolutions = argv[3];
    int res = int((*resolutions)-48);
    //std:: cout << "\nres" << res;

```

```

gdcmm::StreamImageWriter theStreamWriter;

std::ofstream of;
of.open( outfilename, std::ios::out | std::ios::binary );
theStreamWriter.SetStream(of);

if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;

uint16_t firstTag1 = 0xfffe;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize1];
memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
//memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
gdcmm_assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize1);
of.flush();
gdcmm_assert( of );

return 0;
}

```

14.67 Fake_Image_Using_Stream_Image_Writer.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmmReader.h"
#include "gdcmmMediaStorage.h"
#include "gdcmmWriter.h"
#include "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmAttribute.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"
#include "gdcmmTransferSyntax.h"
#include "gdcmmUIDGenerator.h"
#include "gdcmmAnonymizer.h"
#include "gdcmmStreamImageWriter.h"
#include "gdcmmImageHelper.h"
#include "gdcmmTrace.h"

int main(int, char *[])
{
    char * buffer = new char[ 256 * 256 *3 ];
    // *p = (uint8_t*)buffer;
    char * p = buffer;

    gdcmm::Trace::DebugOn();
    gdcmm::Trace::WarningOn();

    for(int row = 0; row < 256; ++row)
    {
        for(int col = 0; col < 256; ++col)
            //for(int b = 0; b < 256; ++b)
            {
                *p++ = 255;
                *p++ = 0;
            }
    }
}

```



```

        *p++ = 0;
    }
}

gdcmm::Writer w;
gdcmm::File &file = w.GetFile();
gdcmm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );

gdcmm::UIDGenerator uid;
gdcmm::DataElement de( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcmm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcmm::DataElement de1( gdcmm::Tag(0x8,0x16) );
de1.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage ms( gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
de1.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( de1 );

const char mystr[] = "RGB";
gdcmm::DataElement de2( gdcmm::Tag(0x28,0x04) );
//de.SetTag(gdcmm::Tag(0x28,0x04));
de2.SetVR( gdcmm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcmm::Attribute<0x0028,0x0010> row = {256};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col = {256};
ds.Insert( col.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel
ds.Insert( at1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0006> at4 = {0};
ds.Insert( at4.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0103> at5 = {0};
ds.Insert( at5.GetAsDataElement() );

//de.SetTag(gdcmm::Tag(0x7fe0,0x0010));
//ds.Insert(de);

gdcmm::StreamImageWriter theStreamWriter;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;

gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
el2.SetValue(1);
gdcmm::DataElement rfn = el2.GetAsDataElement(); //rfn ---> reference frame number
rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el;
el.SetValue(1,0);
el.SetValue(1,1);
gdcmm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper left col/row
ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el1;

```

```

    el1.SetValue(col1,0);
    el1.SetValue(row1,1);
    gdcm::DataElement brr = el1.GetAsDataElement();
    brr.SetTag( gdcm::Tag(0x0048,0x0202) );           //brr --> bottom right col/row

    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert( ulr );
    nds.Insert( brr );

    sq->AddItem(it);

    gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert(des);

    theStreamWriter.SetFile(file);

    std::ofstream of;
    of.open( "output.dcm", std::ios::out | std::ios::binary );
    theStreamWriter.SetStream(of);

    if (!theStreamWriter.CanWriteFile()){
        delete [] buffer;
        std::cout << "Not able to write";
        return 0; //this means that the file was unwritable, period.
        //very similar to a ReadImageInformation failure
    }
    else
        std::cout << "\nable to read";

    if (!theStreamWriter.WriteImageInformation()){
        std::cerr << "unable to write image information" << std::endl;
        delete [] buffer;
        return 1; //the CanWrite function should prevent getting here, else,
        //that's a test failure
    }

    std::vector<unsigned int> extent =
        gdcm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];

    std::cout << xmax << ymax << zmax;

    if (xmax == 0 || ymax == 0)
    {
        std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
        return 0;
    }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially. So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
        for (y = 0; y < ymax; y += ychunk){
            nexty = y + ychunk;
            if (nexty > ymax) nexty = ymax;
            theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
            unsigned long len = theStreamWriter.DefineProperBufferLength();
            std::cout << "\n" << len;
            char* finalBuffer = new char[len];
            memcpy(finalBuffer, &(buffer[prevLen]), len);
            std::cout << "\nable to write";
            if (!theStreamWriter.Write(finalBuffer, len)){
                std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z << std::endl;
                delete [] buffer;
                delete [] finalBuffer;
                return 1;
            }
            delete [] finalBuffer;

```

```

        prevLen += len;
    }
    delete buffer;

    uint16_t firstTag1 = 0xfffe;
    uint16_t secondTag1 = 0xe0dd;
    uint32_t thirdTag1 = 0x00000000;
    //uint16_t fourthTag1 = 0xffff;
    const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
    char* tmpBuffer2 = new char[theBufferSize1];
    memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
    memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
    //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
    gdcmm_assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    gdcmm_assert( of );

    return 0;
}

```

14.68 FixBrokenJ2K.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfFragments.h"
#include "gdcmmFile.h"

// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvua-fixed2-j2k.dcm

/*
* This program attempts to fix a broken J2K/DICOM:
* It contains 2 bugs:
* 1. The first 8 bytes seems to be random bytes: remove them
* 2. YCC is set to 1, while image is grayscale need to set it back to 0
*
* Ref:
* It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
* "ProScan-2000". The Ukrainian manufacturer developed own digital radiographic unit and it is
* compatible with software from "ProScan-2000".
* Information found in DICOM file is:
*
* (0008,0070) LO [ZAO "Renthenprom" (JSC Rentgenprom) ] # 36,1 Manufacturer
* (0018,1020) LO [2.13.1.7] # 8,1-n Software Version(s)
*
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )

```

```

    {
        return 1;
    }

    gdcmm::File &file = reader.GetFile();
    const gdcmm::DataElement &pixeldata0 = file.GetDataSet().GetDataElement( gdcmm::Tag(0x7fe0,0x0010) );
    const gdcmm::SequenceOfFragments *sqf = pixeldata0.GetSequenceOfFragments();
    if( !sqf )
    {
        return 1;
    }
    const gdcmm::Fragment &frag0 = sqf->GetFragment(0);

    gdcmm::ByteValue *bv = const_cast<gdcmm::ByteValue*>(frag0.GetByteValue());
    char *ptr = (char*)bv->GetVoidPointer();
    size_t len = bv->GetLength();

    static const unsigned char sig[] = {0,0,0,0,0x6A,0x70,0x32,0x63};
    if( memcmp(ptr, sig, sizeof(sig)) != 0 )
    {
        std::cerr << "magic random signature not found" << std::endl;
        return 1;
    }

    // Apparently the flag to enable a color transform on 3 color components is set in
    // the COD marker. (YCC is byte[6] in the COD marker)
    // we need to disable this flag;
    char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
    if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
        // found start of COD
        if( cod_marker[6+2] == 1 )
        {
            // Change in place:
            *((char*)cod_marker + 6+2) = 0;
            // Prepare a new DataElement:
            gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0x0010) );
            pixeldata.SetVR( gdcmm::VR::OB );
            gdcmm::SmartPointer<gdcmm::SequenceOfFragments> sq = new gdcmm::SequenceOfFragments;

            gdcmm::Fragment frag;
            // remove 8 first bytes:
            frag.SetByteValue( ptr + 8, (uint32_t)(len - 8) );
            sq->AddFragment( frag );
            pixeldata.SetValue( *sq );
            file.GetDataSet().Replace( pixeldata );
        }
        else
        {
            return 1;
        }
    }
    else
    {
        std::cerr << "COD not found" << (int)cod_marker[0] << std::endl;
        return 1;
    }

    gdcmm::Writer writer;
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    writer.CheckFileMetaInformationOff();
    if( !writer.Write() )
    {
        std::cerr << "Could not write" << std::endl;
    }

    // paranoid check:
    gdcmm::ImageReader ireader;
    ireader.SetFileName( outfilename );
    if( !ireader.Read() )
    {
        std::cerr << "file written is still not valid, please report" << std::endl;
        return 1;
    }

    return 0;
}

```

14.69 FixJAIBugJPEGLS.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"

#include <fstream>

#include "gdcm_charls.h"

/*
 * This small example should show how one can handle the famous JAI-JPEGLS bug
 * It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit Little
 * Endian. One can use `gdcmconv --jpegl's` to recompress properly
 *
 * References:
 * http://charls.codeplex.com/discussions/230307?ProjectName=charls
 * http://charls.codeplex.com/workitem/7297
 * http://www.dcm4che.org/jira/browse/DCM-442
 * http://www.dcm4che.org/jira/browse/DCMEE-1144
 * http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
 *
 * Explanation of the issue:
 *
 * Seems, the error is in the calculation of the default values for thresholds T1,
 * T2, T3, in particular min(MAXVAL, 4095) is not applied in
 *
 * FACTOR = (min(MAXVAL, 4095) + 128)/256
 *
 * as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::FileMetaInformation::SetSourceApplicationEntityTitle( "FixJAIBugJPEGLS" );

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::Image &image = reader.GetImage();
    //unsigned long len = image.GetBufferLength();
    const gdcm::DataElement &in =
        reader.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
    const gdcm::SequenceOfFragments *sf = in.GetSequenceOfFragments();
    if( !sf )
    {
        std::cerr << "No pixel data (or not encapsulated)" << std::endl;
        return 1;
    }
    const unsigned int *dims = image.GetDimensions();
    if ( sf->GetNumberOfFragments() != dims[2] )
    {
        std::cerr << "Unsupported" << std::endl;
        return 1;
    }
}

```

```

    }

// unsigned long totalLen = sf->ComputeByteLength();
std::vector<unsigned char> rgbyteOutall;
for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
{
    const gdcmm::Fragment &frag = sf->GetFragment(i);
    if( frag.IsEmpty() ) return 1;
    const gdcmm::ByteValue *bv = frag.GetByteValue();
    if( !bv ) return 1;
    unsigned long totalLen = bv->GetLength();

    std::vector<char> vbuffer;
    vbuffer.resize( totalLen );
    char *buffer = vbuffer.data();
    bv->GetBuffer(buffer, totalLen);
    const unsigned char* pbyteCompressed0 = (const unsigned char*)buffer;
    while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
    {
        totalLen--;
    }

    JlsParameters metadata;
    char errorMsg[256+1]={'\0'};
    if (JpegLsReadHeader(buffer, totalLen, &metadata, errorMsg) != charls::ApiResult::OK)
    {
        std::cerr << "Can't parse jpegls: " << errorMsg << std::endl;
        return 1;
    }

    std::cout << metadata.width << std::endl;
    std::cout << metadata.height << std::endl;
    std::cout << metadata.bitsPerSample << std::endl;

    gdcmm::PixelFormat const &pf = image.GetPixelFormat();
    std::cout << pf << std::endl;

    // http://charls.codeplex.com/discussions/230307?ProjectName=charls
    unsigned char marker_lse_13[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0x1F, 0xFF,
        0x00, 0x22, // T1 = 34
        0x00, 0x83, // T2 = 131
        0x02, 0x24, // T3 = 548
        0x00, 0x40
    };

    unsigned char marker_lse_14[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0x3F, 0xFF,
        0x00, 0x42, // T1 = 66
        0x01, 0x03, // T2 = 259
        0x04, 0x44, // T3 = 1092
        0x00, 0x40
    };

    unsigned char marker_lse_15[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0x7F, 0xFF,
        0x00, 0x82, // T1 = 130
        0x02, 0x03, // T2 = 515
        0x08, 0x84, // T3 = 2180
        0x00, 0x40
    };

    unsigned char marker_lse_16[] = {
        0xFF, 0xF8, 0x00, 0x0D,
        0x01,
        0xFF, 0xFF,
        0x01, 0x02, // T1 = 258
        0x04, 0x03, // T2 = 1027
        0x11, 0x04, // T3 = 4356
        0x00, 0x40
    };

    const unsigned char *marker_lse = nullptr;
    switch( metadata.bitsPerSample )
    {

```

```

    case 13:
        marker_lse = marker_lse_13;
        break;
    case 14:
        marker_lse = marker_lse_14;
        break;
    case 15:
        marker_lse = marker_lse_15;
        break;
    case 16:
        marker_lse = marker_lse_16;
        break;
    }
    if( !marker_lse )
    {
        std::cerr « "Can't handle: " « metadata.bitsPerSample « std::endl;
        return 1;
    }

    // FIXME: One should recompute the value for 0x0F
    vbuffer.insert( vbuffer.begin() + 0x0F, marker_lse, marker_lse+15);

#ifdef 0
    std::ofstream of( "/tmp/d.jls", std::ios::binary );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif

    const char *pbyteCompressed = vbuffer.data();
    size_t cbyteCompressed = vbuffer.size(); // updated length

    JlsParameters params;
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params, nullptr);

    std::vector<unsigned char> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height * params.width * ((params.bitsPerSample + 7)
        / 8) * params.components);

    CharlsApiResultType result =
        JpegLsDecode(rgbyteOut.data(), rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params, errorMsg );
    if (result != charls::ApiResult::OK)
    {
        std::cerr « "Could not patch JAI-JPEGLS: " « errorMsg « std::endl;
        return 1;
    }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end() );
}

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcm::VR::OW );
pixeldata.SetByteValue( (char*)rgbyteOutall.data(), (uint32_t)rgbyteOutall.size() );

// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );
reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);

gdcm::Writer writer;
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
writer.Write();

std::cout « "Success !" « std::endl;

return 0;
}

```

14.70 FixOrientation.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmFile.h"
#include "gdcmOrientation.h"
#include "gdcmAttribute.h"

// Very simple orientation changer, fix invalid dataset
int main(int argc, char* argv[] )
{
    // assume AXIAL input for now
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    const double axial[] = { 1,0,0, 0,1,0 };
    (void)axial;
    const double coronal[] = { 0,0,1, 1,0,0 };
    (void)coronal;
    const double sagittal[] = { 0,1,0, 0,0,1 };
    (void)sagittal;
    gdcm::Attribute<0x0020,0x0032> at1; // IPP
    (void)at1;
    gdcm::Attribute<0x0020,0x0037> at2; // IOP
    (void)at2;

    gdcm::File & f = reader.GetFile();
    gdcm::DataSet & ds = f.GetDataSet();
    at1.SetFromDataSet( ds );
    #if 0
    at2.SetFromDataSet( ds );
    const double * iop = at2.GetValues();
    if( !std::equal(iop, iop + 6, axial ) )
    {
        gdcm::Orientation::OrientationType type = gdcm::Orientation::GetType ( iop );
        std::cerr << "Wrong orientation: " << gdcm::Orientation::GetLabel( type ) << std::endl;
        return 1;
    }
    at2.SetValues( sagittal );
    ds.Replace( at2.GetAsDataElement() );
    #endif

    // for sagittal: swap element 0 & 2
    const double tmp0 = at1.GetValue(0);
    const double tmp2 = at1.GetValue(2);
    (void)tmp2;
    //at1.SetValue(tmp2, 0);
    //at1.SetValue(tmp0, 2);
    at1.SetValue( - tmp0 );
    ds.Replace( at1.GetAsDataElement() );

    gdcm::Writer writer;
    writer.SetFile( f );
    writer.SetFileName( outfile );
    if ( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```


14.71 GenAllVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFileExplicitFilter.h"

#include <cstdlib>
#include <cstring>

gdcm::Tag FindTagFromVR(gdcm::Dict const &dict, gdcm::VR const &vr)
{
    using gdcm::Dict;
    Dict::ConstIterator beg = dict.Begin();
    Dict::ConstIterator end = dict.End();
    Dict::ConstIterator it;
    for( it = beg; it != end; ++it)
    {
        const gdcm::Tag &t = it->first;
        const gdcm::DictEntry &de = it->second;
        const gdcm::VR &vr_de = de.GetVR();
        if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
        {
            return t;
        }
    }
    return gdcm::Tag(0xffff,0xffff);
}

struct rnd_gen {
    rnd_gen(char const* r = "abcdefghijklmnopqrstuvwxyz0123456789")
        : range(r), len(std::strlen(r)) { }

    char operator ()() const {
        return range[static_cast<std::size_t>(std::rand() * (1.0 / ((double)RAND_MAX + 1.0)) * (double)len)];
    }
private:
    char const* range;
    std::size_t len;
};

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];
    static const gdcm::Global &g = gdcm::Global::GetInstance();
    static const gdcm::Dicts &dicts = g.GetDicts();
    static const gdcm::Dict &pubdict = dicts.GetPublicDict();

```

```

using gdcm::VR;
using gdcm::Tag;

gdcm::Writer w;

gdcm::File &f = w.GetFile();
gdcm::DataSet &ds = f.GetDataSet();

gdcm::FileExplicitFilter fef;
//fef.SetChangePrivateTags( true );
fef.SetFile( w.GetFile() );
if( !fef.Change() )
{
    std::cerr << "Failed to change" << std::endl;
    return 1;
}

gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
sq->SetLengthToUndefined();

// gdcm::DummyValueGenerator dv;

const std::size_t len = 10;
char ss[len+1];
ss[len] = '\0';

const char owner_str[] = "GDCM CONFORMANCE TESTS";
gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
owner.SetVR( gdcm::VR::LO );

// Create an item
gdcm::Item it;
it.SetVLToUndefined();
gdcm::DataSet &nds = it.GetNestedDataSet();
// nds.Insert(owner);
// nds.Insert(de);

// Insert sequence into data set
gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
des.SetVR(gdcm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

// avoid INVALID = 0
for(int i = 1; i < 27; ++i)
{
    VR vr = (VR::VRType)(1LL << i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
    {
        gdcm_assert( t != Tag(0xffff,0xffff) );
        gdcm::DataElement de( t );
        std::generate_n(ss, len, rnd_gen());
        de.SetVR( vr );
        de.SetByteValue( ss, (uint32_t)std::strlen( ss ) );
        nds.Insert( de );
    }
}
sq->AddItem(it);

// Make sure to override any UID stuff
gdcm::UIDGenerator uid;
gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
ds.Insert( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
ds.Insert( de );

gdcm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );

```

```

fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

14.72 GenFakeIdentifyFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmGlobal.h"
#include "gdcmmDummyValueGenerator.h"
#include "gdcmmMediaStorage.h"
#include "gdcmmWriter.h"
#include "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfItems.h"
#include "gdcmmAttribute.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"
#include "gdcmmDict.h"
#include "gdcmmDictEntry.h"
#include "gdcmmDicts.h"
#include "gdcmmTransferSyntax.h"
#include "gdcmmUIDGenerator.h"
#include "gdcmmAnonymizer.h"

#include <cstdlib>
#include <cstring>

gdcmm::DataElement CreateFakeElement(gdcmm::Tag const &tag, bool toremove)
{
    static const gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();
    static size_t countglobal = 0;
    static std::vector<gdcmm::Tag> balcptags =
        gdcmm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
    size_t count = countglobal % balcptags.size();

    const gdcmm::DictEntry &dictentry = pubdict.GetDictEntry(tag);

    gdcmm::DataElement de;
    de.SetTag( tag );
    using gdcmm::VR;
    const VR &vr = dictentry.GetVR();
    //if( vr != VR::INVALID )
    if( vr.IsDual() )
    {
        if( vr == VR::US_SS )
        {
            de.SetVR( VR::US );
        }
        else if( vr == VR::US_SS_OW )
        {
            de.SetVR( VR::OW );
        }
        else if( vr == VR::OB_OW )
        {

```

```

        de.SetVR( VR::OB );
    }
}
else
{
    de.SetVR( vr );
}
const char str[] = "BasicApplicationLevelConfidentialityProfileAttributes";
const char safe[] = "This is safe to keep";
if( de.GetVR() != VR::SQ )
{
    if( toremove )
        de.SetByteValue( str, (uint32_t)strlen(str) );
    else
        de.SetByteValue( safe, (uint32_t)strlen(safe) );
}
else
{
    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    gdcmm::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    gdcmm::assert(de.GetVR() == gdcmm::VR::SQ);
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new gdcmm::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
    de.SetVLToUndefined();
    //ds.Insert(de);

    if( !toremove )
    {
        nds.Insert( CreateFakeElement( balcptags[count], true ) );
        countglobal++;
    }
    else
    {
        gdcmm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no reason to be 'anonymized'...
        nds.Insert( at1.GetAsDataElement() );
        gdcmm::Attribute<0x000a,0x0000> at2 = { 0 };
        nds.Insert( at2.GetAsDataElement() );
    }
    sq->AddItem(it);
}
return de;
}

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    using gdcmm::Tag;
    using gdcmm::VR;
    const char *outfilename = argv[1];

    std::vector<gdcmm::Tag> balcptags =
        gdcmm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();

    gdcmm::Writer w;
    gdcmm::File &f = w.GetFile();
    gdcmm::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdcmm::Tag>::const_iterator it = balcptags.begin();
    for(; it != balcptags.end(); ++it)
    {
        ds.Insert( CreateFakeElement( *it, true ) );
    }

    // Add attribute that do NOT need to be anonymized:
    static const gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();

    using gdcmm::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();

```

```

for( dictit != pubdict.End(); ++dictit)
{
    const gdcmm::Tag &dicttag = dictit->first;
    if( dicttag == Tag(0x6e65,0x6146) ) break;
    //const gdcmm::DictEntry &dictentry = dictit->second;
    ds.Insert( CreateFakeElement( dicttag, false ) );
}
ds.Remove( gdcmm::Tag(0x400,0x500) );
ds.Remove( gdcmm::Tag(0x12,0x62) );
ds.Remove( gdcmm::Tag(0x12,0x63) );

// Make sure to override any UID stuff
gdcmm::UIDGenerator uid;
gdcmm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
//ds.Insert( de );
ds.Replace( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcmm::MediaStorage ms( gdcmm::MediaStorage::RawDataStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
ds.Replace( de ); // replace !

gdcmm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ImplicitVRLittleEndian );
fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

14.73 GenLongSeqs.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfItems.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"

/*
 * This example is used to generate the file:
 *
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * We need to make sure that we can store numerous Item in a SQ
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])

```

```

{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Create a Sequence
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );

    size_t nitens = 1000;
    nitens += std::numeric_limits<uint32_t>::max();
    for(unsigned int idx = 0; idx < nitens; ++idx)
    {
        // Create a dataelement
        //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
        //de.SetByteValue(ptr, ptr_len);
        //de.SetVR( gdcm::VR::OB );

        // Create an item
        gdcm::Item it;
        it.SetVLToUndefined();
        //gdcm::DataSet &nds = it.GetNestedDataSet();
        //nds.Insert(owner);
        //nds.Insert(de);

        sq->AddItem(it);
    }

    // Insert sequence into data set
    gdcm::DataElement des( gdcm::Tag(0x4d4d, 0x1001) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert(owner);
    ds.Insert(des);

    gdcm::Writer w;
    w.SetFile( file );
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfile );
    if( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

14.74 GenSeqs.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmConformanceTests/SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most cases, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * Deflated syntax was used in this case since this synthetic example can be
 * nicely compressed using this transfer syntax.
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    //const unsigned int nitems = 1000;
    const unsigned int ptr_len = 42; /*94967296 / nitems; */
    //gdcm_assert( ptr_len == 42949672 );
    char *ptr = new char[ptr_len];
    memset(ptr,0,ptr_len);

    // Create a Sequence
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );

    for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
    {
        // Create a dataelement
        gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
        de.SetByteValue(ptr, ptr_len);
        de.SetVR( gdcm::VR::OB );

        // Create an item
        gdcm::Item it;
        it.SetVLToUndefined();
        gdcm::DataSet &nds = it.GetNestedDataSet();
        nds.Insert(owner);
        nds.Insert(de);

        sq->AddItem(it);
    }

    // Insert sequence into data set
    gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );

```

```

des.SetVR(gdcm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

gdcm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

14.75 GenerateStandardSOPClasses.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/

#include "gdcmDefs.h"
#include "gdcmUIDs.h"
#include "gdcmGlobal.h"
#include "gdcmMediaStorage.h"
#include "gdcmSOPClassUIDToIOD.h"

int main(int , char *[])
{
    using gdcm::MediaStorage;
    gdcm::Global& g = gdcm::Global::GetInstance();
    if (!g.LoadResourcesFiles() )
    {
        std::cerr << "Could not LoadResourcesFiles" << std::endl;
        return 1;
    }

    const gdcm::Defs &defs = g.GetDefs();

    int ret = 0;

    //std::cout << "Table B.5-1 STANDARD SOP CLASSES" << std::endl;
    std::cout << "SOP Class Name,SOP Class UID,IOD Specification (defined in PS 3.3)" << std::endl;

    gdcm::MediaStorage::MSType mst;
    for ( mst = gdcm::MediaStorage::MediaStorageDirectoryStorage; mst < gdcm::MediaStorage::MS_END;
        mst = (gdcm::MediaStorage::MSType)(mst + 1) )
    {
        const char *iod = defs.GetIODNameFromMediaStorage(mst);
        gdcm::UIDs uid;
        uid.SetFromUID( gdcm::MediaStorage::GetMSString(mst) /*mst.GetString()*/ );
        if ( iod )
        {
            const char *iod_ref = gdcm::SOPClassUIDToIOD::GetIOD(uid);
            if ( iod_ref )
            {
                std::string iod_ref_str = iod_ref;
                //iod_ref_str += " IOD Modules";
                //if( iod_ref_str != iod )
            }
        }
    }
}

```



```

        //std::cout << "UID: " << uid << " ";
        std::cout << "''" << uid.GetName() << "''" << "," << "''" << uid.GetString() << "''" << "," << "''" << iod << "''" << std::endl;
        //std::cout << "Incompatible IODs: [" << iod << "]" versus ref= [" << iod_ref_str << "]" << std::endl;
        ++ret;
    }
}
}

return 0;
}

```

14.76 GetJPEGSamplePrecision.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This example is a little helper to detect the famous SIEMENS JPEG lossless compressed image
* where DICOM is declared as:
*
* (0028,0100) US 16                                # 2,1 Bits Allocated
* (0028,0101) US 12                                # 2,1 Bits Stored
* (0028,0102) US 11                                # 2,1 High Bit
* (0028,0103) US 0                                  # 2,1 Pixel Representation
*
* But where JPEG is:
*
*     JPEG_SOF_Parameters:
*         SamplePrecision = 16
*         nLines = 192
*         nSamplesPerLine = 192
*         nComponentsInFrame = 1
*         component 0
*             ComponentIdentifier = 1
*             HorizontalSamplingFactor = 1
*             VerticalSamplingFactor = 1
*             QuantizationTableDestinationSelector = 0
*
* This case is valid. One simply has to use the 16bits jpeg decoder to decode the 12bits stored image.
* This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
*
* The main return 0 (no error) when the file read is actually a potential problem. At the end of the main
* function, the jpeg stream is stored in the filename specified as second argument
*/

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.jpg" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );

```

```

if( !reader.Read() )
{
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}

// The output of gdcm::Reader is a gdcm::File
const gdcm::File &file = reader.GetFile();
const gdcm::Image &image = reader.GetImage();

const gdcm::TransferSyntax &ts = file.GetHeader().GetDataSetTransferSyntax();

if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14 && ts != gdcm::TransferSyntax::JPEGLosslessProcess14_1 )
{
    std::cerr << "Input is not a lossless JPEG" << std::endl;
    return 1;
}

// the dataset is the the set of element we are interested in:
const gdcm::DataSet &ds = file.GetDataSet();

const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
const gdcm::DataElement& pdde = ds.GetDataElement( rawTag );
const gdcm::SequenceOfFragments *sf = pdde.GetSequenceOfFragments();
if( sf )
{
    std::ofstream output(outfilename, std::ios::binary);
    sf->WriteBuffer(output);
}
else
{
    std::cerr << "Error" << std::endl;
    return 1;
}

gdcm::JPEGCodec jpeg;
std::ifstream is(outfilename, std::ios::binary);
gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8 ); // let's pretend it's a 8bits jpeg
jpeg.SetPixelFormat( pf );
gdcm::TransferSyntax ts_jpg;
bool b = jpeg.GetHeaderInfo( is, ts_jpg );
if( !b )
{
    return 1;
}

//jpeg.Print( std::cout );
if( jpeg.GetPixelFormat().GetBitsAllocated() != image.GetPixelFormat().GetBitsAllocated()
|| jpeg.GetPixelFormat().GetBitsStored() != image.GetPixelFormat().GetBitsStored() )
{
    std::cerr << "There is a mismatch in between DICOM declared Pixel Format and Sample Precision used in the JPEG stream" <<
        std::endl;
    return 0;
}

std::cout << jpeg.GetPixelFormat() << std::endl;
std::cout << image.GetPixelFormat() << std::endl;

return 1;
}

```

14.77 GetSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"

```

```

#include "gdcmAttribute.h"

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max );

int main(int argc, char* argv[] )
{
    // Controllo del numero di argomenti introdotti da riga di comando
    if( argc < 2 )
    {
        std::cerr « "Usage: " « std::endl;
        std::cerr « argv[0] « " inputImageFile " « std::endl;
        return EXIT_FAILURE;
    }

    unsigned int x_min = 1;
    unsigned int y_min = 1;
    unsigned int x_max = 1;
    unsigned int y_max = 1;

    if( Region ( argv[1], &x_min, &y_min, &x_max, &y_max ) )
    {
        std::cout « "x_min = " « x_min « std::endl;
        std::cout « "y_min = " « y_min « std::endl;
        std::cout « "x_max = " « x_max « std::endl;
        std::cout « "y_max = " « y_max « std::endl;
    }

    else
    {
        std::cout « "no\n";
    }
}

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max )
{
    gdcm::Reader reader;
    reader.SetFileName( nomefile );
    if( !reader.Read() )
    {
        std::cerr « "Could not read: " « nomefile « std::endl;
        return false;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    gdcm::Tag tsqr(0x0018,0x6011);
    if( !ds.FindDataElement( tsqr ) )
    {
        return false;
    }

    const gdcm::DataElement &sqr = ds.GetDataElement( tsqr );
    //std::cout « sqr « std::endl;
    const gdcm::SequenceOfItems *sqi = sqr.GetValueAsSQ();
    if( !sqi || !sqi->GetNumberOfItems() )
    {
        return false;
    }
    //std::cout « sqi « std::endl;

    const gdcm::Item &item = sqi->GetItem(1);
    //std::cout « item « std::endl;
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout « nestedds « std::endl;

    gdcm::Tag tX0(0x0018,0x6018);
    gdcm::Tag tY0(0x0018,0x601a);
    gdcm::Tag tX1(0x0018,0x601c);
    gdcm::Tag tY1(0x0018,0x601e);

    if( (!nestedds.FindDataElement( tX0 ))||(!nestedds.FindDataElement( tY0 ))||(!nestedds.FindDataElement( tX1
    )))||(!nestedds.FindDataElement( tY1 )) )
    {
        return false;
    }

    const gdcm::DataElement& deX0 = nestedds.GetDataElement( tX0 );
    const gdcm::DataElement& deY0 = nestedds.GetDataElement( tY0 );

```

```

const gdcm::DataElement& deX1 = nestedds.GetDataElement( tX1 );
const gdcm::DataElement& deY1 = nestedds.GetDataElement( tY1 );
//std::cout << deX0 << std::endl << deY0 << std::endl << deX1 << std::endl << deY1 << std::endl;

//const gdcm::ByteValue *bvX0 = deX0.GetByteValue();
//const gdcm::ByteValue *bvY0 = deY0.GetByteValue();
//const gdcm::ByteValue *bvX1 = deX1.GetByteValue();
//const gdcm::ByteValue *bvY1 = deY1.GetByteValue();
//std::cout << bvX0 << std::endl << bvY0 << std::endl << bvX1 << std::endl << bvY1 << std::endl;

gdcm::Attribute<0x0018,0x6018> atX0;
gdcm::Attribute<0x0018,0x601a> atY0;
gdcm::Attribute<0x0018,0x601c> atX1;
gdcm::Attribute<0x0018,0x601e> atY1;
atX0.SetFromDataElement( deX0 );
atY0.SetFromDataElement( deY0 );
atX1.SetFromDataElement( deX1 );
atY1.SetFromDataElement( deY1 );
uint32_t X0 = atX0.GetValue();
uint32_t Y0 = atY0.GetValue();
uint32_t X1 = atX1.GetValue();
uint32_t Y1 = atY1.GetValue();
std::cout << X0 << std::endl << Y0 << std::endl << X1 << std::endl << Y1 << std::endl;

*X_min = static_cast<unsigned int>(X0);
*Y_min = static_cast<unsigned int>(Y0);
*X_max = static_cast<unsigned int>(X1);
*Y_max = static_cast<unsigned int>(Y1);

//std::cout << "X_min = " << *X_min << std::endl;
//std::cout << "Y_min = " << *Y_min << std::endl;
//std::cout << "X_max = " << *X_max << std::endl;
//std::cout << "Y_max = " << *Y_max << std::endl;

return true;
}

```

14.78 GetSubSequenceData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <sstream>
#include <string>

#include <map>

/*
* This example will extract the Movie from the private group of
* GEMS_Ultrasound_MovieGroup_001 See Attribute
* (7fe1,60,GEMS_Ultrasound_MovieGroup_001)
*
* The output file will be stored in `outvid.dcm' as
* MultiframeGrayscaleByteSecondaryCaptureImageStorage
*/

using namespace gdcm;
static bool processgroup(Item & item3, std::string const & outfilename)
{

```

```

// Item &item3 = sqi3->GetItem(1);
DataSet &subds3 = item3.GetNestedDataSet();

const PrivateTag tseq6(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
if( !subds3.FindDataElement( tseq6 ) ) return true;
const DataElement& seq6 = subds3.GetDataElement( tseq6 );
SmartPointer<SequenceOfItems> sqi6 = seq6.GetValueAsSQ();
size_t ni6= sqi6->GetNumberOfItems();
gdcmm_assert( sqi6->GetNumberOfItems() >= 1 );
const PrivateTag tseq7(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001");
int dimx = 0, dimy = 0;
for( size_t i6 = 1; i6 <= ni6; ++i6 )
{
    Item &item6 = sqi6->GetItem(i6);
    DataSet &subds6 = item6.GetNestedDataSet();

    if( subds6.FindDataElement( tseq7 ) )
    {
        Element<VR::SL, VM::VM4> el;
        el.SetFromDataElement( subds6.GetDataElement( tseq7 ) );
        dimx = el.GetValue(0);
        dimy = el.GetValue(1);
        std::cout << "Dims= " << dimx << " " << dimy << std::endl;
    }
}

const PrivateTag tseq3(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
if( !subds3.FindDataElement( tseq3 ) ) return true;
const DataElement& seq3 = subds3.GetDataElement( tseq3 );

// std::cout << seq3 << std::endl;

SmartPointer<SequenceOfItems> sqi4 = seq3.GetValueAsSQ();
size_t ni4= sqi4->GetNumberOfItems();
gdcmm_assert( sqi4->GetNumberOfItems() >= 1 );
const PrivateTag tseq8(0x7fe1,0x37,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq4(0x7fe1,0x43,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq5(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");

std::vector<char> imbuffer;
int dimz = 0;
for( size_t i4 = 1; i4 <= ni4; ++i4 )
{
    Item &item4 = sqi4->GetItem(i4);
    DataSet &subds4 = item4.GetNestedDataSet();

    if( !subds4.FindDataElement( tseq8 ) ) return true;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL, VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    std::cout << "ldimz: " << ldimz.GetValue() << std::endl;
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return true;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return true;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );

    // std::cout << seq4 << std::endl;
    // std::cout << seq5 << std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
    Element<VR::FD, VM::VM1_n> el0;
    el0.SetFromDataElement( seq4 );
    std::cout << "TimeStamp (" << el0.GetLength() << "): ";
    // Seems like the 3D volumes is split into chunks of max 100 frames...
    gdcmm_assert( ldimz.GetValue() == el0.GetLength() );
    for( unsigned long i = 0; i < el0.GetLength(); ++i ) {
        if(i) std::cout << ", ";
        std::cout << el0.GetValue(i);
    }
    std::cout << std::endl;
}
#endif
{
    std::ofstream out( "/tmp/mo4", std::ios::binary );
    out.write( bv4->GetPointer(), bv4->GetLength() );
    out.close();
}
#endif
const ByteValue *bv5 = seq5.GetByteValue();
#endif

```

```

    {
        std::ofstream out( "tmp/mo5", std::ios::binary );
        out.write( bv5->GetPointer(), bv5->GetLength());
        out.close();
    }
#endif

    std::cout << bv5->GetLength() << std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->GetPointer() + bv5->GetLength() );
}
DataElement fakedata;
fakedata.SetByteValue( imbuffer.data(), (uint32_t)imbuffer.size() );

gdcmm::SmartPointer<gdcmm::Image> im = new gdcmm::Image;
im->SetNumberOfDimensions( 3 );

im->SetDimension(0, dimx );
im->SetDimension(1, dimy );
im->SetDimension(2, dimz );
size_t l1 = imbuffer.size();
(void)l1;
size_t l2 = im->GetBufferLength();
(void)l2;
gdcmm::assert( im->GetBufferLength() == imbuffer.size() );
im->SetPhotometricInterpretation( gdcmm::PhotometricInterpretation::MONOCHROME2 );

im->SetDataElement( fakedata );

gdcmm::ImageWriter w;
w.SetImage( *im );
DataSet &dataset = w.GetFile().GetDataSet();

gdcmm::UIDGenerator uid;
gdcmm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, (uint32_t)strlen(u) );
//ds.Insert( de );
dataset.Replace( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcmm::MediaStorage ms(
    gdcmm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
dataset.Replace( de ); // replace !

w.SetFileName( outfilename.c_str() );
if( !w.Write() )
{
    return false;
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
    gdcmm::assert( sqi->GetNumberOfItems() == 1 );
    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq1(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds.FindDataElement( tseq1 ) ) return 1;

```

```

const DataElement& seq1 = subds.GetDataElement( tseq1 );

SmartPointer<SequenceOfItems> sqi2 = seq1.GetValueAsSQ();
gdcm_assert( sqi2->GetNumberOfItems() == 1 );
//int n = sqi2->GetNumberOfItems();
int index = 1;
Item &item2 = sqi2->GetItem(index);
DataSet &subds2 = item2.GetNestedDataSet();

const PrivateTag tseq2(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");

if( !subds2.FindDataElement( tseq2 ) ) return 1;
const DataElement& seq2 = subds2.GetDataElement( tseq2 );

//  std::cout << seq2 << std::endl;

SmartPointer<SequenceOfItems> sqi3 = seq2.GetValueAsSQ();
size_t ni3 = sqi3->GetNumberOfItems(); (void)ni3;
gdcm_assert( sqi3->GetNumberOfItems() >= 1 );
std::cout << " #Groups = " << sqi3->GetNumberOfItems() << std::endl;
for( SequenceOfItems::SizeType i = 1; i <= sqi3->GetNumberOfItems(); ++i ) {
    Item &item3 = sqi3->GetItem(i);
    std::ostringstream os;
    os << "outvid";
    os << i;
    os << ".dcm";
    processgroup(item3, os.str());
}

return 0;
}

```

14.79 HelloVizWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the image reader:
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
}

```

```

// If we reach here, we know for sure 2 things:
// 1. It is a valid DICOM
// 2. And it contains an Image !

// The output of superclass gdcm::Reader is a gdcm::File
//gdcm::File &file = reader.GetFile();

// The other output of gdcm::ImageReader is a gdcm::Image
const gdcm::Image &image = reader.GetImage();

// Let's get some property from the image:
unsigned int ndim = image.GetNumberOfDimensions();
// Dimensions of the image:
const unsigned int *dims = image.GetDimensions();
// Origin
const double *origin = image.GetOrigin();
const gdcm::PhotometricInterpretation &pi = image.GetPhotometricInterpretation();
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Dim(" << i << "): " << dims[i] << std::endl;
}
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Origin(" << i << "): " << origin[i] << std::endl;
}
std::cout << "PhotometricInterpretation: " << pi << std::endl;

// Write the modified DataSet back to disk
gdcm::ImageWriter writer;
writer.SetImage( image );
writer.SetFileName( outfilename );
//writer.SetFile( file ); // We purposely NOT copy the meta information from the input
// file, and instead only pass the image
if( !writer.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

14.80 HelloWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example is ... guess what this is for :)
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the reader:

```



```

gdcmm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}

// If we reach here, we know for sure only 1 thing:
// It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
// (Maybe, it's NOT a Dicom image -could be a DicomDIR, a RTSTRUCT, etc-)

// The output of gdcmm::Reader is a gdcmm::File
gdcmm::File &file = reader.GetFile();

// the dataset is the the set of element we are interested in:
gdcmm::DataSet &ds = file.GetDataSet();

// Construct a static(*) type for Image Comments :
gdcmm::Attribute<0x0020,0x4000> imagecomments;
imagecomments.SetValue( "Hello, World !" );

// Now replace the Image Comments from the dataset with our:
ds.Replace( imagecomments.GetAsDataElement() );

// Write the modified DataSet back to disk
gdcmm::Writer writer;
writer.CheckFileMetaInformationOff(); // Do not attempt to reconstruct the file meta to preserve the file
// as close to the original as possible.
writer.SetFileName( outfilename );
writer.SetFile( file );
if( !writer.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
 * The compiler is deducing those values from the template arguments of the class.
 */

```

14.81 LargeVRDSExplicit.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmFileExplicitFilter.h"
#include "gdcmmSequenceOfItems.h"

bool interpolate(const double * pts, size_t npts, std::vector<double> &out )
{
    out.clear();
    for(size_t i = 0; i < 2*npts; ++i )
    {
        const size_t j = i / 2;
        if( i % 2 )
        {
            if( j != npts - 1 )
            {

```

```

        gdcmm_assert( 3*j+5 < 3*npts );
        const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
        const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
        const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
        out.push_back( midpointx );
        out.push_back( midpointy );
        out.push_back( midpointz );
    }
}
else
{
    gdcmm_assert( j < npts );
    out.push_back( pts[3*j+0] );
    out.push_back( pts[3*j+1] );
    out.push_back( pts[3*j+2] );
}
}
gdcmm_assert( out.size() == 2 * npts * 3 - 3 );
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();

    gdcmm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( changeprivatetags );
    fef.SetFile( reader.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change: " << filename << std::endl;
        return 1;
    }

    // (3006,0039) SQ (Sequence with undefined length #=4)    # u/l, 1 ROIContourSequence
    gdcmm::Tag tag(0x3006,0x0039);

    const gdcmm::DataElement &roicsq = ds.GetDataElement( tag );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = roicsq.GetValueAsSQ();
    //sqi->SetNumberOfItems( 1 );
    const gdcmm::Item &item = sqi->GetItem(1); // Item start at #1
    const gdcmm::DataSet& nestedds = item.GetNestedDataSet();

    gdcmm::Tag tcsq(0x3006,0x0040);
    if( !nestedds.FindDataElement( tcsq ) )
    {
        return 0;
    }
    const gdcmm::DataElement& csq = nestedds.GetDataElement( tcsq );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi2 = csq.GetValueAsSQ();
    if( !sqi2 || !sqi2->GetNumberOfItems() )
    {
        return 0;
    }
    //unsigned int nititems = sqi2->GetNumberOfItems();
    gdcmm::Item &item2 = sqi2->GetItem(1); // Item start at #1

    gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //item2.SetVLToUndefined();
    //std::cout << nestedds2 << std::endl;
    // (3006,0050) DS [43.57636\65.52504\ -10.0\46.043102\62.564945\ -10.0\49.126537\60.714... # 398,48 ContourData
    gdcmm::Tag tcontourdata(0x3006,0x0050);
    const gdcmm::DataElement &contourdata = nestedds2.GetDataElement( tcontourdata );
    //std::cout << contourdata << std::endl;

    //const gdcmm::ByteValue *bv = contourdata.GetByteValue();

```

```

gdcmm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcmm::Attribute<0x3006,0x0050> at;
at.SetFromDataElement( contourdata );
const double* pts = at.GetValues();
unsigned int npts = at.GetNumberOfValues() / 3;

std::vector<double> out( pts, pts + npts * 3 );
std::vector<double> out2;

//const unsigned int niter = 7;
const unsigned int niter = 8;
for( unsigned int i = 0; i < niter; ++i)
{
    //bool b =
    interpolate(out.data(), out.size() / 3, out2);
    //const double *pout = &out[0];
    out = out2;
    out2.clear();
}
gdcmm_assert( out.size() % 3 == 0 );

gdcmm::Attribute<0x3006,0x0050> at_interpolate;
at_interpolate.SetNumberOfValues( (unsigned int)(out.size() / 3) );
at_interpolate.SetValues( out.data(), (uint32_t)out.size() );

ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
nestedds2.Replace( at_interpolate.GetAsDataElement() );
nestedds2.Replace( ncontourpoints.GetAsDataElement() );

//gdcmm_assert(0);

// Let's take item one and subdivide it

gdcmm::TransferSyntax ts = gdcmm::TransferSyntax::ImplicitVRLittleEndian;
ts = gdcmm::TransferSyntax::ExplicitVRLittleEndian;

gdcmm::FileMetaInformation &fmi = file.GetHeader();
const char *tsuid = gdcmm::TransferSyntax::GetTSString( ts );
// const char * is ok since padding is \0 anyway...
gdcmm::DataElement de( gdcmm::Tag(0x0002,0x0010) );
de.SetByteValue( tsuid, (uint32_t)strlen(tsuid) );
de.SetVR( gdcmm::Attribute<0x0002, 0x0010>::GetVR() );
fmi.Replace( de );
fmi.Remove( gdcmm::Tag(0x0002,0x0012) ); // will be regenerated
fmi.Remove( gdcmm::Tag(0x0002,0x0013) ); // ' ', ' '
fmi.SetDataSetTransferSyntax(ts);

gdcmm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

14.82 MakeTemplate.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmFileAnonymizer.h"

```

```

#include "gdcmReader.h"
#include "gdcmWriter.h"

int main(int argc, char *argv[])
{
    if( argc < 3 ) return 1;
    const char* filename = argv[1];
    const char* outfilename = argv[2];

    //gdcm::Trace::DebugOn();

    // Remove Pixel Data element:
    gdcm::FileAnonymizer fa;
    fa.SetInputFileName( filename );
    fa.SetOutputFileName( outfilename );

    fa.Empty( gdcm::Tag(0x7fe0,0x10) );
    // cannot replace in-place DICOM header:
    //fa.Replace( gdcm::Tag(0x2,0x2), "1.2.840.10008.5.1.4.1.1.7" );

    if( !fa.Write() )
    {
        std::cerr << "impossible to remove Pixel Data attribute" << std::endl;
        return 1;
    }

    // Update the DICOM Header:
    gdcm::Reader reader;
    reader.SetFileName( outfilename );
    if( !reader.Read() )
    {
        std::cerr << "could not read back" << std::endl;
        return 1;
    }

    gdcm::File & file = reader.GetFile();
    gdcm::FileMetaInformation &fmi = file.GetHeader();
    gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
    ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;
    fmi.SetDataSetTransferSyntax(ts);

    gdcm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfilename ); // warning overwrite file !
    if( !writer.Write() )
    {
        std::cerr << "could not write back" << std::endl;
        return 1;
    }

    return 0;
}

```

14.83 MergeTwoFiles.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This example will show how one can read in two DICOM files, use the dataset
* from file1 and use image from file2 to save it in a 3rd file.
*
* Eg:
* MergeTwoFiles gdcmData/012345.002.050.dcm gdcmData/test.acr merge.dcm
*/

#include "gdcmReader.h"

```

```

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *file1 = argv[1];
    const char *file2 = argv[2];
    const char *file3 = argv[3];

    // Read file1
    gdcm::ImageReader reader1;
    reader1.SetFileName( file1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    // Read file2
    gdcm::ImageReader reader2;
    reader2.SetFileName( file2 );
    if( !reader2.Read() )
    {
        return 1;
    }

    // Ok now let's take the DataSet from file1 and the Image from file2
    // Warning: if file2 is -for example- a Secondary Capture Storage, then it has no
    // Image Orientation (Patient) thus any Image Orientation (Patient) from file1
    // will be discarded...

    // let's be fancy. In case reader2 contains explicit, but reader1 is implicit
    // we would rather see an implicit output
    if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax() == gdcm::TransferSyntax::ImplicitVRLittleEndian )
    {
        reader2.GetImage().SetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
    }

    gdcm::ImageWriter writer;
    writer.SetFileName( file3 );
    writer.SetFile( reader1.GetFile() );
    // ImageWriter will always use all of gdcm::Image information an override anything wrong from
    // reader1.GetFile(), including the Transfer Syntax
    writer.SetImage( reader2.GetImage() );

    gdcm::DataSet &ds = reader1.GetFile().GetDataSet();

    // Make sure that SOPInstanceUID are different
    // Simply removing it is sufficient as gdcm::ImageWriter will generate one by default
    // if not found.
    ds.Remove( gdcm::Tag(0x0008,0x0018) );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

14.84 MrProtocol.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

PURPOSE. See the above copyright notice for more information.

```

===== */
/*
 *
 */

/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN ###
ulVersion = 0xbee332
tSequenceFileName = "%SiemensSeq%\fl_fq_shphs"
tProtocolName = "flash+AF8-100+AF8-through-plane+AF8-V"
tReferenceImage0 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004658"
tReferenceImage1 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004635"
tReferenceImage2 = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004683"
ucScanRegionPosValid = 0x1
sProtConsistencyInfo.tBaselineString = "N4_VB11A_LATEST_20031004"
sProtConsistencyInfo.flNominalB0 = 1.494
sProtConsistencyInfo.flGMax = 22
sProtConsistencyInfo.flRiseTime = 10
sGRADSPEC.sEddyCompensationX.aflAmplitude[0] = 0.0141111
sGRADSPEC.sEddyCompensationX.aflAmplitude[1] = 0.057038
sGRADSPEC.sEddyCompensationX.aflAmplitude[2] = -0.00986504
sGRADSPEC.sEddyCompensationX.aflAmplitude[3] = 0.00247627
sGRADSPEC.sEddyCompensationX.aflAmplitude[4] = 0.0026377
sGRADSPEC.sEddyCompensationX.aflTimeConstant[0] = 1.53826
sGRADSPEC.sEddyCompensationX.aflTimeConstant[1] = 0.746617
sGRADSPEC.sEddyCompensationX.aflTimeConstant[2] = 0.339236
sGRADSPEC.sEddyCompensationX.aflTimeConstant[3] = 0.0309809
sGRADSPEC.sEddyCompensationX.aflTimeConstant[4] = 0.00067694
sGRADSPEC.sEddyCompensationY.aflAmplitude[0] = 0.0156411
sGRADSPEC.sEddyCompensationY.aflAmplitude[1] = 0.0440623
sGRADSPEC.sEddyCompensationY.aflAmplitude[2] = -0.00782663
sGRADSPEC.sEddyCompensationY.aflAmplitude[3] = 0.00186828
sGRADSPEC.sEddyCompensationY.aflAmplitude[4] = 0.00154504
sGRADSPEC.sEddyCompensationY.aflTimeConstant[0] = 1.47145
sGRADSPEC.sEddyCompensationY.aflTimeConstant[1] = 0.750538
sGRADSPEC.sEddyCompensationY.aflTimeConstant[2] = 0.339397
sGRADSPEC.sEddyCompensationY.aflTimeConstant[3] = 0.0312962
sGRADSPEC.sEddyCompensationY.aflTimeConstant[4] = 0.000895133
sGRADSPEC.sEddyCompensationZ.aflAmplitude[0] = 0.00618504
sGRADSPEC.sEddyCompensationZ.aflAmplitude[1] = 0.00313121
sGRADSPEC.sEddyCompensationZ.aflAmplitude[2] = 0.000289346
sGRADSPEC.sEddyCompensationZ.aflAmplitude[3] = -0.00019677
sGRADSPEC.sEddyCompensationZ.aflAmplitude[4] = 7.66445e-005
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[0] = 3.37462
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[1] = 0.999351
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[2] = 0.0174646
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[3] = 0.0110094
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[4] = 0.00199922
sGRADSPEC.bEddyCompensationValid = 1
sGRADSPEC.sB0CompensationX.aflAmplitude[0] = 0.307474
sGRADSPEC.sB0CompensationX.aflAmplitude[1] = 0.029337
sGRADSPEC.sB0CompensationX.aflAmplitude[2] = -0.187118
sGRADSPEC.sB0CompensationX.aflTimeConstant[0] = 0.98583
sGRADSPEC.sB0CompensationX.aflTimeConstant[1] = 0.0308443
sGRADSPEC.sB0CompensationX.aflTimeConstant[2] = 0.000466792
sGRADSPEC.sB0CompensationY.aflAmplitude[0] = 0.365257
sGRADSPEC.sB0CompensationY.aflAmplitude[1] = -0.318647
sGRADSPEC.sB0CompensationY.aflAmplitude[2] = -0.0118978
sGRADSPEC.sB0CompensationY.aflTimeConstant[0] = 0.61535
sGRADSPEC.sB0CompensationY.aflTimeConstant[1] = 0.488831
sGRADSPEC.sB0CompensationY.aflTimeConstant[2] = 0.00199991
sGRADSPEC.sB0CompensationZ.aflAmplitude[0] = -0.44647
sGRADSPEC.sB0CompensationZ.aflAmplitude[1] = -0.0455154
sGRADSPEC.sB0CompensationZ.aflAmplitude[2] = -0.0304901
sGRADSPEC.sB0CompensationZ.aflTimeConstant[0] = 0.959231
sGRADSPEC.sB0CompensationZ.aflTimeConstant[1] = 0.0720189
sGRADSPEC.sB0CompensationZ.aflTimeConstant[2] = 0.00190141
sGRADSPEC.bB0CompensationValid = 1
sGRADSPEC.sCrossTermCompensationXY.aflAmplitude[0] = 0.00105046
sGRADSPEC.sCrossTermCompensationXY.aflTimeConstant[0] = 0.842014
sGRADSPEC.sCrossTermCompensationXZ.aflAmplitude[0] = -0.00150189
sGRADSPEC.sCrossTermCompensationXZ.aflTimeConstant[0] = 0.736169
sGRADSPEC.sCrossTermCompensationYX.aflAmplitude[0] = -5.5278e-005
sGRADSPEC.sCrossTermCompensationYX.aflTimeConstant[0] = 0.228697
sGRADSPEC.sCrossTermCompensationYZ.aflAmplitude[0] = 0.000307999
sGRADSPEC.sCrossTermCompensationYZ.aflTimeConstant[0] = 1.19431
sGRADSPEC.sCrossTermCompensationZX.aflAmplitude[0] = -0.000286868
sGRADSPEC.sCrossTermCompensationZX.aflTimeConstant[0] = 0.665979
sGRADSPEC.sCrossTermCompensationZY.aflAmplitude[0] = 0.000355175

```

```

sGRADSPEC.sCrossTermCompensationZY.afTimeConstant[0] = 0.844189
sGRADSPEC.bCrossTermCompensationValid = 1
sGRADSPEC.lOffsetX = 25
sGRADSPEC.lOffsetY = 84
sGRADSPEC.lOffsetZ = 47
sGRADSPEC.bOffsetValid = 1
sGRADSPEC.lDelayX = 12
sGRADSPEC.lDelayY = 11
sGRADSPEC.lDelayZ = 9
sGRADSPEC.bDelayValid = 1
sGRADSPEC.flSensitivityX = 0.000264087
sGRADSPEC.flSensitivityY = 0.000272009
sGRADSPEC.flSensitivityZ = 0.000272677
sGRADSPEC.bSensitivityValid = 1
sGRADSPEC.alShimCurrent[0] = 183
sGRADSPEC.alShimCurrent[1] = -25
sGRADSPEC.alShimCurrent[2] = -85
sGRADSPEC.alShimCurrent[3] = 378
sGRADSPEC.alShimCurrent[4] = 82
sGRADSPEC.bShimCurrentValid = 1
sGRADSPEC.ucMode = 0x2
sTXSPEC.asNucleusInfo[0].tNucleus = "1H"
sTXSPEC.asNucleusInfo[0].lFrequency = 63684693
sTXSPEC.asNucleusInfo[0].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[0].flReferenceAmplitude = 359.734
sTXSPEC.asNucleusInfo[0].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[0].flAmplitudeCorrection = 1
sTXSPEC.asNucleusInfo[0].bAmplitudeCorrectionValid = 1
sTXSPEC.asNucleusInfo[1].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[1].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[1].bAmplitudeCorrectionValid = 1
sTXSPEC.aRFPULSE[0].tName = "03GreFCE"
sTXSPEC.aRFPULSE[0].bAmplitudeValid = 0x1
sTXSPEC.aRFPULSE[0].flAmplitude = 147.095
sTXSPEC.aRFPULSE[1].tName = "02GreFCE"
sTXSPEC.aRFPULSE[1].bAmplitudeValid = 0x1
sTXSPEC.aRFPULSE[1].flAmplitude = 147.095
sTXSPEC.aRFPULSE[2].tName = "01GreFCE"
sTXSPEC.aRFPULSE[2].bAmplitudeValid = 0x1
sTXSPEC.aRFPULSE[2].flAmplitude = 147.095
sTXSPEC.lNoOfTraPulses = 3
sTXSPEC.lBTB1ParallelCapacity = 2
sTXSPEC.lBTB1SerialCapacity = 24
sTXSPEC.lBTB2ParallelCapacity = 2
sTXSPEC.lBTB2SerialCapacity = 26
sTXSPEC.bBTBValid = 1
sTXSPEC.flKDynMagnitudeMin = 0.5
sTXSPEC.flKDynMagnitudeMax = 1.5
sTXSPEC.flKDynMagnitudeClipLow = 0.96
sTXSPEC.flKDynMagnitudeClipHigh = 1.04
sTXSPEC.flKDynPhaseMax = 0.698132
sTXSPEC.flKDynPhaseClip = 0.174533
sTXSPEC.bKDynValid = 1
sTXSPEC.ucRFPulseType = 0x1
sTXSPEC.ucExcitMode = 0x1
sTXSPEC.ucSimultaneousExcitation = 0x1
sRXSPEC.lGain = 1
sRXSPEC.bGainValid = 1
sRXSPEC.aFFT_SCALE[0].lRxChannel = 1
sRXSPEC.aFFT_SCALE[0].flFactor = 1.06857
sRXSPEC.aFFT_SCALE[0].bValid = 1
sRXSPEC.aFFT_SCALE[1].lRxChannel = 2
sRXSPEC.aFFT_SCALE[1].flFactor = 1.07454
sRXSPEC.aFFT_SCALE[1].bValid = 1
sRXSPEC.aFFT_SCALE[2].lRxChannel = 3
sRXSPEC.aFFT_SCALE[2].flFactor = 1.06622
sRXSPEC.aFFT_SCALE[2].bValid = 1
sRXSPEC.aFFT_SCALE[3].lRxChannel = 4
sRXSPEC.aFFT_SCALE[3].flFactor = 1.06524
sRXSPEC.aFFT_SCALE[3].bValid = 1
sRXSPEC.aFFT_SCALE[4].lRxChannel = 5
sRXSPEC.aFFT_SCALE[4].flFactor = 0.982692
sRXSPEC.aFFT_SCALE[4].bValid = 1
sRXSPEC.aFFT_SCALE[5].lRxChannel = 6
sRXSPEC.aFFT_SCALE[5].flFactor = 0.988603
sRXSPEC.aFFT_SCALE[5].bValid = 1
sRXSPEC.aFFT_SCALE[6].lRxChannel = 7
sRXSPEC.aFFT_SCALE[6].flFactor = 0.981538
sRXSPEC.aFFT_SCALE[6].bValid = 1
sRXSPEC.aFFT_SCALE[7].lRxChannel = 8
sRXSPEC.aFFT_SCALE[7].flFactor = 1.00856

```

```

sRXSPEC.aFFT_SCALE[7].bValid          = 1
sRXSPEC.bVariCapVoltagesValid         = 1
sRXSPEC.alDwellTime[0]                = 8500
sAdjFreSpec.ulMode                    = 0x1
sAdjFreSpec.ucAdjWithBC               = 0x1
sAdjTraSpec.ucAdjWithBC              = 0x1
sAdjShimSpec.ulMode                   = 0x1
sAdjShimSpec.ucAdjWithBC             = 0x1
sAdjWatSupSpec.ulMode                 = 0x1
sAdjWatSupSpec.ucAdjWithBC           = 0x1
alTR[0]                              = 37000
lContrasts                           = 1
alTE[0]                              = 4000
acFlowComp[0]                        = 1
lCombinedEchoes                      = 1
sSliceArray.asSlice[0].sPosition.dSag = 35.31199581
sSliceArray.asSlice[0].sPosition.dCor = -8.387765754
sSliceArray.asSlice[0].sPosition.dTra = -23.13178296
sSliceArray.asSlice[0].sNormal.dSag   = 0.771051253
sSliceArray.asSlice[0].sNormal.dCor   = 0.5863890019
sSliceArray.asSlice[0].sNormal.dTra   = -0.2482496801
sSliceArray.asSlice[0].dThickness     = 6
sSliceArray.asSlice[0].dPhaseFOV      = 187.5
sSliceArray.asSlice[0].dReadoutFOV    = 250
sSliceArray.lSize                     = 1
sSliceArray.lSag                      = 1
sSliceArray.lConc                     = 1
sSliceArray.ucMode                    = 0x1
sSliceArray.sTSat.dThickness          = 40
sSliceArray.sTSat.dGap                = 10
sGroupArray.asGroup[0].nSize          = 1
sGroupArray.asGroup[0].dDistFact      = 0.2
sGroupArray.anMember[1]               = -1
sGroupArray.lSize                     = 1
sGroupArray.sPSat.dThickness          = 50
sGroupArray.sPSat.dGap                = 10
sAutoAlign.dAAMatrix[0]               = 1
sAutoAlign.dAAMatrix[5]               = 1
sAutoAlign.dAAMatrix[10]              = 1
sAutoAlign.dAAMatrix[15]              = 1
sNavigatorPara.ucRespComp             = 0x4
sPrepPulses.ucFatSat                  = 0x4
sPrepPulses.ucWaterSat                = 0x4
sPrepPulses.ucInversion                = 0x4
sPrepPulses.ucSatRecovery              = 0x1
sPrepPulses.ucFatSatMode              = 0x2
sKSpace.lBaseResolution                = 256
sKSpace.lPhaseEncodingLines           = 192
sKSpace.dPhaseResolution               = 1
sKSpace.lPartitions                    = 32
sKSpace.lImagesPerSlab                = 32
sKSpace.dSliceResolution               = 1
sKSpace.ucPhasePartialFourier          = 0x10
sKSpace.ucSlicePartialFourier          = 0x10
sKSpace.ucAveragingMode                = 0x2
sKSpace.ucMultiSliceMode               = 0x1
sKSpace.ucDimension                    = 0x2
sKSpace.ucAsymmetricEchoAllowed        = 0x1
sKSpace.unReordering                  = 0x1
sFastImaging.lEPIFactor                = 1
sFastImaging.lTurboFactor              = 1
sFastImaging.lSegments                 = 3
sFastImaging.ulEnableRFSpoiling        = 0x1
sPhysioImaging.lSignal1                = 2
sPhysioImaging.lMethod1                = 2
sPhysioImaging.lSignal2                = 1
sPhysioImaging.lMethod2                = 1
sPhysioImaging.lPhases                 = 21
sPhysioImaging.lRetroGatedImages       = 16
sPhysioImaging.sPhysioECG.lScanWindow = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5

```



```

sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio = 0.3
sSpecPara.lPhaseCyclingType = 1
sSpecPara.lPhaseEncodingType = 1
sSpecPara.lRFExcitationBandwidth = 1
sSpecPara.ucRemoveOversampling = 0x1
sSpecPara.lDecouplingType = 1
sSpecPara.lNOEType = 1
sSpecPara.lExcitationType = 1
sSpecPara.lSpectralSuppression = 1
sDiffusion.ulMode = 0x1
sAngio.sFlowArray.asElm[0].nVelocity = 100
sAngio.sFlowArray.asElm[0].nDir = 0x4
sAngio.sFlowArray.lSize = 1
sAngio.ucPCFlowMode = 0x2
sAngio.ucTOFInflow = 0x4
sAngio.ucRephasedImage = 0x1
sAngio.ucPhaseImage = 0x1
sEllipticalFilter.ucMode = 0x1
sPat.lAccelFactPE = 1
sPat.lAccelFact3D = 1
sPat.ucPATMode = 0x1
sPat.ucRefScanMode = 0x1
ucAutoMovie = 0x1
ucDisableChangeStoreImages = 0x1
ucReconstructionMode = 0x1
ucPHAPSMODE = 0x1
ucDixon = 0x1
lAverages = 2
adFlipAngleDegree[0] = 30
lScanTimeSec = 103
lTotalScanTimeSec = 112
dRefSNR = 165404.1473
dRefSNR_VOI = 165404.1473
tdefaultEVAProt = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1

```

```

sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid = 1
### ASCCONV END ###
*/

/*
* Table of equivalence:
*
ulVersion = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"

#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    gdcm::CSAHeader csa;
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

    if( ds.FindDataElement( t2 ) )
    {
        csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
        //csa.Print( std::cout );
    }

    if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
        return 1;
    }
}

```

```

const gdcmm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
//std::cout << csael << std::endl;

const gdcmm::ByteValue *bv = csael.GetByteValue();
if( !bv )
{
    return 1;
}
std::string str(bv->GetPointer(), bv->GetLength());
std::istringstream is(str);
std::string s;
typedef std::map< std::string, std::string > MyMapType;
MyMapType mymap;
while( std::getline(is, s) )
{
    std::string::size_type pos = s.find( '=' );
    if( pos != std::string::npos )
    {
        std::string sub1 = s.substr(0, pos);
        sub1.erase( sub1.find_last_not_of(' ') + 1);
        std::string sub2 = s.substr(pos+1); // skip the '=' char
        sub2.erase( 0, sub2.find_first_not_of(' '));
        //std::cout << sub1 << std::endl;
        mymap.insert( MyMapType::value_type(sub1, sub2) );
    }
    else
    {
        // ### ASCCONV BEGIN ###
        // ### ASCCONV END ###
    }
}
const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
const gdcmm::CSAHeaderDict &csadict = gdcmm::Global::GetInstance().GetDicts().GetCSAHeaderDict();
const gdcmm::CSAHeaderDictEntry &fourier = csadict.GetCSAHeaderDictEntry( fourierstr );
std::cout << fourier << std::endl;
MyMapType::const_iterator it = mymap.find ( fourierstr );
if( it == mymap.end() ) return 1;
//std::cout << it->second << std::endl;
const std::string &partial_fourier = it->second;
if( partial_fourier == "0x1" )
{
    std::cout << "partial fourier is 4/8" << std::endl;
}
else if( partial_fourier == "0x2" )
{
    std::cout << "partial fourier is 5/8" << std::endl;
}
else if( partial_fourier == "0x4" )
{
    std::cout << "partial fourier is 6/8" << std::endl;
}
else if( partial_fourier == "0x8" )
{
    std::cout << "partial fourier is 7/8" << std::endl;
}
else if( partial_fourier == "0x10" )
{
    std::cout << "partial fourier is 8/8" << std::endl;
}
else
{
    std::cerr << "Impossible: " << partial_fourier << std::endl;
    return 1;
}
/*
This is the Flip Angle:
adFlipAngleDegree[0]          = 30

```

One can find it also in the protocol:

```

...
    <ParamFuncutor."T1mapFuncutor">
    {
        <Class> "T1mapFuncutor@IceImagePostProcFuncutors"

        <ParamBool."EXECUTE"> { }
        <ParamDouble."Flip1_deg"> { <Precision> 16 14.7378520000000000 }
    }
...

*/
// Below is an attempt to play with the CSAHeader dict:

```

```

#if 0
const char gspec[] = "sGRADSPEC.flSensitivityX";
it = mymap.find( gspec );
if( it == mymap.end() ) return 1;
const std::string &dummy = it->second;
std::cout << dummy << std::endl;

const gdcm::CSAHeaderDictEntry &csaentry = csadict.GetCSAHeaderDictEntry( gspec );
std::cout << csaentry << std::endl;
#endif

/*
sSliceArray.ucMode -- should be in (1, 2, 4)
enum SeriesMode
{
    ASCENDING = 0x01,
    DESCENDING = 0x02,
    INTERLEAVED = 0x04
};
*/
const char sliceorderstr[] = "sSliceArray.ucMode";
const gdcm::CSAHeaderDictEntry &sliceorder = csadict.GetCSAHeaderDictEntry( sliceorderstr );
std::cout << sliceorder << std::endl;

it = mymap.find( sliceorderstr );
if( it == mymap.end() ) return 1;
const std::string &slice_order = it->second;
if( slice_order == "0x1" )
{
    std::cout << "slice_order: ASCENDING" << std::endl;
}
else if( slice_order == "0x2" )
{
    std::cout << "slice_order: DESCENDING" << std::endl;
}
else if( slice_order == "0x4" )
{
    std::cout << "slice_order: INTERLEAVED" << std::endl;
}
else
{
    std::cerr << "Impossible: " << slice_order << std::endl;
    return 1;
}

gdcm::MrProtocol mrprot;
if( csa.GetMrProtocol(ds, mrprot) )
{
    std::cout << mrprot << std::endl;
}

return 0;
}

```

14.85 PrintLUT.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"

```

```

#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the image reader:
    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    const gdcm::Image &image = reader.GetImage();

    const gdcm::LookupTable &lut = image.GetLUT();
    lut.Print( std::cout );

    return 0;
}

```

14.86 PublicDict.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
    const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all knowledge !
    const gdcm::Dicts &dicts = g.GetDicts();
    const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6

    //std::cout << pub << std::endl;

    // 3 different ways to access the same information

    // 1. From the public dict only:
    gdcm::Tag patient_name(0x10,0x10);
    const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
    std::cout << entry1 << std::endl;

    // 2. From all dicts:
    const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
    std::cout << entry2 << std::endl;

    // 3. This solution is the most flexible solution as you can request using the same
    // API either a public tag or a private tag
    const char *strowner = nullptr;
    const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);

```

```

std::cout << entry3 << std::endl;

// Private attributes:

// try with a private tag now:
const gdcm::PrivateTag &private_tag = gdcm::CSAHeader::GetCSAImageHeaderInfoTag();
//std::cout << private_tag << std::endl;
const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.GetOwner());
std::cout << entry4 << std::endl;

// Let's pretend that private lookup is on 0x10xx elements:
gdcm::PrivateTag dummy = private_tag;
dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.GetOwner());
std::cout << entry5 << std::endl;

return 0;
}

```

14.87 QIDO-RS.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcm.JSON.h"

/*
 * Simple QIDO-RS round-trip to test implementation of gdcm::JSON
 * See Sup166 for details
 */
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::JSON json;
    json.PrettyPrintOn();
    std::stringstream ss;
    const gdcm::File & f = reader.GetFile();
    json.Code( f.GetDataSet(), ss);

    std::cout << ss.str() << std::endl;

    gdcm::Writer w;
    gdcm::File & ff = w.GetFile();
    ff.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
    if( !json.Decode(ss, ff.GetDataSet() ) )
    {
        std::cerr << "Could not decode" << std::endl;
        return 1;
    }
    w.SetFileName( "/tmp/debug.dcm" );
    if( !w.Write() ) return 1;

    return 0;
}

```

14.88 ReadAndDumpDICOMDIR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 * Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmlReader.h"
#include "gdcmlMediaStorage.h"

typedef std::set<gdcml::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];

    gdcml::Reader reader;
    reader.SetFileName( filename);
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;

    gdcml::File &file = reader.GetFile();
    gdcml::DataSet &ds = file.GetDataSet();
    gdcml::FileMetaInformation &fmi = file.GetHeader();

    gdcml::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcml::MediaStorage::MediaStorageDirectoryStorage )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }

    if( fmi.FindDataElement( gdcml::Tag( 0x0002, 0x0002) ) )
    {
        strm.str("");
        fmi.GetDataElement( gdcml::Tag( 0x0002, 0x0002) ).GetValue().Print(strm);
    }
    else
    {
        std::cerr << " Media Storage Sop Class UID not present" << std::endl;
    }

    //TODO il faut trimer strm.str() avant la comparaison au cas ou...
    if( "1.2.840.10008.1.3.10"!=strm.str() )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }

    ConstIterator it = ds.GetDES().begin();
    for( ; it != ds.GetDES().end(); ++it )
    {
        if( it->GetTag()==gdcml::Tag( 0x0004, 0x1220) )
        {
            const gdcml::DataElement &de = (*it);
            // ne pas utiliser GetSequenceOfItems pour extraire les items
            gdcml::SmartPointer<gdcml::SequenceOfItems> sqi =de.GetValueAsSQ();

```

```

unsigned int itemused = 1;
while (itemused <= sqi->GetNumberOfItems())

{
    strm.str("");

    if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

    //TODO il faut trimer strm.str() avant la comparaison
    while((strm.str()=="PATIENT")||((strm.str()=="PATIENT ")))
    {
        std::cout << strm.str() << std::endl;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0010)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0010)).GetValue().Print(strm);
        std::cout << "PATIENT NAME : " << strm.str() << std::endl;

        //PATIENT ID
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0020)).GetValue().Print(strm);
        std::cout << "PATIENT ID : " << strm.str() << std::endl;

        /*ADD TAG TO READ HERE*/
        std::cout << "===== " << std::endl;
        itemused++;
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

        //TODO il faut trimer strm.str() avant la comparaison
        while((strm.str()=="STUDY")||((strm.str()=="STUDY ")))
        {
            std::cout << " " << strm.str() << std::endl;
            //UID
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000d)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
            std::cout << " STUDY UID : " << strm.str() << std::endl;

            //STUDY DATE
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0020)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
            std::cout << " STUDY DATE : " << strm.str() << std::endl;

            //STUDY DESCRIPTION
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x1030)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x1030)).GetValue().Print(strm);
            std::cout << " STUDY DESCRIPTION : " << strm.str() << std::endl;

            /*ADD TAG TO READ HERE*/
            std::cout << " " << "===== " << std::endl;

            itemused++;
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

            //TODO il faut trimer strm.str() avant la comparaison
            while((strm.str()=="SERIES")||((strm.str()=="SERIES ")))
            {
                std::cout << " " << strm.str() << std::endl;
                strm.str("");
                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000e)))
                    sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
                std::cout << " SERIE UID" << strm.str() << std::endl;

                //SERIE MODALITY
                strm.str("");
                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0060)))
                    sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
                std::cout << " SERIE MODALITY" << strm.str() << std::endl;

                //SERIE DESCRIPTION
                strm.str("");
                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x103e)))
                    sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x103e)).GetValue().Print(strm);

```



```

std::cout << "          SERIE DESCRIPTION" << strm.str() << std::endl;

/*ADD TAG TO READ HERE*/

std::cout << "          " << "===== " << std::endl;
itemused++;
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
    sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

//TODO il faut trimer strm.str() avant la comparaison
while ((strm.str()=="IMAGE")||((strm.str()=="IMAGE ")))
    // if(tmp=="IMAGE")
    {
        std::cout << "          " << strm.str() << std::endl;

        //UID
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1511)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1511)).GetValue().Print(strm);
        std::cout << "          IMAGE UID : " << strm.str() << std::endl;

        //PATH de l'image
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1500)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1500)).GetValue().Print(strm);
        std::cout << "          IMAGE PATH : " << strm.str() << std::endl;
        /*ADD TAG TO READ HERE*/

        if(itemused < sqi->GetNumberOfItems())
        {itemused++;
        }else{break;}

        strm.str("");

        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);
        }
    }
    itemused++;
}
}
}
return 0;
}

```

14.89 ReadAndDumpDICOMDIR2.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2017 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This example shows how to read and dump a DICOMDIR File
*
* Thanks:
* Tom Marynowski (lordglub gmail) for contributing the original
* ReadAndDumpDICOMDIR.cxx example
* Mihail Isakov for contributing offset calculation code here:
* https://sourceforge.net/p/gdcm/mailman/gdcm-developers/?viewmonth=201707&viewday=15

```

```

*   Tod Baudais for combining the above and cleaning up this example
*/

#include <string>
#include <unordered_map>
#include <iostream>
#include <memory>

#include "gdcmlReader.h"
#include "gdcmlAttribute.h"
#include "gdcmlDirectory.h"

//=====
//=====

#define TAG_MEDIA_STORAGE_SOP_CLASS_UID 0x0002,0x0002
#define TAG_DIRECTORY_RECORD_SEQUENCE 0x0004,0x1220
#define TAG_DIRECTORY_RECORD_TYPE 0x0004,0x1430
#define TAG_PATIENTS_NAME 0x0010,0x0010
#define TAG_PATIENT_ID 0x0010,0x0020
#define TAG_STUDY_DATE 0x0008,0x0020
#define TAG_STUDY_DESCRIPTION 0x0008,0x1030
#define TAG_MODALITY 0x0008,0x0060
#define TAG_SERIES_DESCRIPTION 0x0008,0x103E
#define TAG_REFERENCED_FILE_ID 0x0004,0x1500
#define TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET 0x0004,0x1420
#define TAG_NEXT_DIRECTORY_RECORD_OFFSET 0x0004,0x1400

//=====
// Some handy utility functions
//=====

std::string left_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(ss.begin(), std::find_if(ss.begin(), ss.end(), std::not1(std::ptr_fun<int, int>(std::isspace))));
    return ss;
}

std::string right_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(std::find_if(ss.rbegin(), ss.rend(), std::not1(std::ptr_fun<int, int>(std::isspace))).base(), ss.end());
    return ss;
}

std::string trim(const std::string &s) {
    return left_trim(right_trim(s));
}

//=====
// This code could be put in a header file somewhere
//=====

class DICOMDIRReader {
public:
    DICOMDIRReader() {}
    DICOMDIRReader(const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader(DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader & operator = (const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader & operator = (DICOMDIRReader &&rhs) = delete;
    virtual ~DICOMDIRReader() {}

public:
    struct Common {
        int64_t child_offset;
        int64_t sibling_offset;
    };

    struct Image: public Common {
        std::string path;
    };

    struct Series: public Common {
        std::string modality;
        std::string description;

        std::vector<std::shared_ptr<Image>> children;
    };

    struct Study: public Common {
        std::string date;
    };

```

```

        std::string description;

        std::vector<std::shared_ptr<Series>> children;
    };

    struct Patient: public Common {
        std::string name;
        std::string id;

        std::vector<std::shared_ptr<Study>> children;
    };

    struct Other: public Common {
    };

    const std::vector<std::shared_ptr<Patient>>& load (const std::string &path);

    const std::vector<std::shared_ptr<Patient>>& patients (void) { return __patients; }

private:

    template <class T>
    std::string get_string (const T &ds, const gdcm::Tag &tag)
    {
        std::stringstream strm;
        if (ds.FindDataElement(tag)) {
            auto &de = ds.GetDataElement(tag);
            if (!de.IsEmpty() && !de.IsUndefinedLength())
                de.GetValue().Print(strm);
        }
        return trim(strm.str());
    }

    template <class P, class C, class O>
    void reassemble_hierarchy (P &parent_offsets, C &child_offsets, O &other_offsets)
    {
        for (auto &parent : parent_offsets) {
            int64_t sibling_offset;
            auto c = child_offsets[parent.second->child_offset];
            if (!c) {
                auto o = other_offsets[parent.second->child_offset];
                if (!o) {
                    continue;
                } else {
                    sibling_offset = o->sibling_offset;
                }
            } else {
                parent.second->children.push_back(c);
                sibling_offset = c->sibling_offset;
            }

            // Get all siblings
            while (sibling_offset) {
                c = child_offsets[sibling_offset];
                if (!c) {
                    auto o = other_offsets[sibling_offset];
                    if (!o) {
                        break;
                    } else {
                        sibling_offset = o->sibling_offset;
                    }
                } else {
                    parent.second->children.push_back(c);
                    sibling_offset = c->sibling_offset;
                }
            }
        }
    }

    std::vector<std::shared_ptr<Patient>> __patients;
};

//=====
// This code could be put in an implementation file somewhere
//=====

const std::vector<std::shared_ptr<DICOMDIRReader::Patient>>& DICOMDIRReader::load (const std::string &path)
{
    __patients.clear();

    //

```

```

// Read the dataset from the DICOMDIR file
//

gdcmm::Reader reader;
reader.SetFileName(path.c_str());
if(!reader.Read()) {
    throw std::runtime_error("Unable to read file");
}

// Retrieve information from file
auto &file = reader.GetFile();
auto &data_set = file.GetDataSet();
auto &file_meta_information = file.GetHeader();

// Retrieve and check the Media Storage class from file
gdcmm::MediaStorage media_storage;
media_storage.SetFromFile(file);
if(media_storage != gdcmm::MediaStorage::MediaStorageDirectoryStorage) {
    throw std::runtime_error("This file is not a DICOMDIR");
}

auto media_storage_sop_class_uid = get_string(file_meta_information,
    gdcmm::Tag(TAG_MEDIA_STORAGE_SOP_CLASS_UID));

// Make sure we have a DICOMDIR file
if(media_storage_sop_class_uid != "1.2.840.10008.1.3.10") {
    throw std::runtime_error("This file is not a DICOMDIR");
}

//
// Offset to first item courtesy of Mihail Isakov
//

gdcmm::VL first_item_offset = 0;
auto it = data_set.Begin();
for(; it != data_set.End() && it->GetTag() != gdcmm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE); ++it) {
    first_item_offset += it->GetLength<gdcmm::ExplicitDataElement>();
}
// Tag (4 bytes)
first_item_offset += it->GetTag().GetLength();
// VR field
first_item_offset += it->GetVR().GetLength();
// VL field
// For Explicit VR: adventitiously VL field length = VR field length,
// for SQ 4 bytes:
// http://dicom.nema.org/medical/dicom/current/output/html/part05.html#table_7.1-1
first_item_offset += it->GetVR().GetLength();

//
// Iterate all data elements
//

// For each item in data set
for(auto data_element : data_set.GetDES()) {

    // Only look at Directory sequence
    if(data_element.GetTag() != gdcmm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE))
        continue;

    auto item_sequence = data_element.GetValueAsSQ();
    auto num_items = item_sequence->GetNumberOfItems();

    //
    // Compute an offset table
    //

    // Start calculation of offset to each item courtesy of Mihail Isakov
    std::vector<int64_t> item_offsets(num_items+1);
    item_offsets[0] = file_meta_information.GetFullLength() + static_cast<int64_t>(first_item_offset);

    //
    // Extract out all of the items
    //

    std::unordered_map<int64_t, std::shared_ptr<Patient>> patient_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Study>> study_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Series>> series_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Image>> image_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Other>> other_offsets;

    for (uint32_t item_index = 1; item_index <= num_items; ++item_index) {

```

```

    auto &item = item_sequence->GetItem(item_index);

    // Add offset for item to offset table
    item_offsets[item_index] = item_offsets[item_index-1] + item.GetLength<gdcm::ExplicitDataElement>();

    // Child offset
    gdcm::Attribute<TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET> child_offset;
    child_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
(TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET)));

    // Sibling offset
    gdcm::Attribute<TAG_NEXT_DIRECTORY_RECORD_OFFSET> sibling_offset;
    sibling_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag (TAG_NEXT_DIRECTORY_RECORD_OFFSET)));

    // Record Type
    auto record_type = trim(get_string(item, gdcm::Tag (TAG_DIRECTORY_RECORD_TYPE)));

    // std::cout << "record_type " << record_type << " at " << item_offsets[item_index-1] << std::endl;
    // std::cout << " child_offset " << child_offset.GetValue() << std::endl;
    // std::cout << " sibling_offset " << sibling_offset.GetValue() << std::endl;

    // Extract patient information
    if (record_type == "PATIENT") {
        auto patient = std::make_shared<Patient>();
        patient->name = get_string(item, gdcm::Tag (TAG_PATIENTS_NAME));
        patient->id = get_string(item, gdcm::Tag (TAG_PATIENT_ID));

        patient->child_offset = child_offset.GetValue();
        patient->sibling_offset = sibling_offset.GetValue();
        patient_offsets[item_offsets[item_index-1]] = patient;

    // Extract study information
    } else if (record_type == "STUDY") {
        auto study = std::make_shared<Study>();
        study->date = get_string(item, gdcm::Tag (TAG_STUDY_DATE));
        study->description = get_string(item, gdcm::Tag (TAG_STUDY_DESCRIPTION));

        study->child_offset = child_offset.GetValue();
        study->sibling_offset = sibling_offset.GetValue();
        study_offsets[item_offsets[item_index-1]] = study;

    // Extract series information
    } else if (record_type == "SERIES") {
        auto series = std::make_shared<Series>();
        series->modality = get_string(item, gdcm::Tag (TAG_MODALITY));
        series->description = get_string(item, gdcm::Tag (TAG_SERIES_DESCRIPTION));

        series->child_offset = child_offset.GetValue();
        series->sibling_offset = sibling_offset.GetValue();
        series_offsets[item_offsets[item_index-1]] = series;

    // Extract image information
    } else if (record_type == "IMAGE") {
        auto image = std::make_shared<Image>();
        image->path = get_string(item, gdcm::Tag (TAG_REFERENCED_FILE_ID));

        image->child_offset = child_offset.GetValue();
        image->sibling_offset = sibling_offset.GetValue();
        image_offsets[item_offsets[item_index-1]] = image;
    } else {
        auto other = std::make_shared<Other>();

        other->child_offset = child_offset.GetValue();
        other->sibling_offset = sibling_offset.GetValue();
        other_offsets[item_offsets[item_index-1]] = other;
    }
}

// Check validity
if (patient_offsets.size() == 0)
    throw std::runtime_error("Unable to find patient record");

reassemble_hierarchy(series_offsets, image_offsets, other_offsets);
reassemble_hierarchy(study_offsets, series_offsets, other_offsets);
reassemble_hierarchy(patient_offsets, study_offsets, other_offsets);

// Set the new root
for (auto &patient : patient_offsets) {
    _patients.push_back(patient.second);
}

```

```

    }

    return __patients;
}

//=====
// Quick test
//=====

int main(int argc, const char *argv[]) {
    DICOMDIRReader reader;

    try {
        if (argc != 2)
            throw std::runtime_error("Wrong number of arguments");

        auto &patients = reader.load(argv[1]);

        for (auto &patient : patients) {

            std::cout << "PATIENT" << std::endl;
            std::cout << "NAME: " << patient->name << std::endl;
            std::cout << "ID: " << patient->id << std::endl;

            int x = 0;
            for (auto &study : patient->children) {
                std::cout << "    STUDY" << std::endl;
                std::cout << "    DESCRIPTION: " << study->description << std::endl;
                std::cout << "    DATE: " << study->date << std::endl;

                for (auto &series : study->children) {
                    x+=1;
                    std::cout << "        SERIES " << x << std::endl;
                    std::cout << "        DESCRIPTION: " << series->description << std::endl;
                    std::cout << "        MODALITY: " << series->modality << std::endl;

                    for (auto &image : series->children) {
                        std::cout << "            IMAGE PATH: " << image->path << std::endl;
                    }
                }
            }
        }
    } catch (...) {
        // TODO handle this
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}

```

14.90 ReadAndPrintAttributes.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmlReader.h"
#include "gdcmlGlobal.h"
#include "gdcmlDicts.h"
#include "gdcmlDict.h"
#include "gdcmlAttribute.h"

```

```

#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Global& g = gdcm::Global::GetInstance();
    const gdcm::Dicts &dictionaries = g.GetDicts();
    const gdcm::Dict &pubdict = dictionaries.GetPublicDict();

    using namespace gdcm;

    // In this example we will show why using name to lookup attribute can be
    // dangerous.
    Tag tPatientName(0x00,0x00);
    //const DictEntry &de1 =
    pubdict.GetDictEntryByName("Patient Name", tPatientName);

    std::cout << "Found: " << tPatientName << std::endl;

    // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
    // has become Patient's Name.

    Tag tPatientsName;
    //const DictEntry &de2 =
    pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

    std::cout << "Found: " << tPatientsName << std::endl;

    // Let's try to read an arbitrary DICOM Attribute:
    Tag tDoseGridScaling;
    //const DictEntry &de3 =
    pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

    std::cout << "Found: " << tDoseGridScaling << std::endl;

    if( ds.FindElement( tDoseGridScaling ) )
    {
        gdcm::StringFilter sf;
        sf.SetFile(file);
        std::cout << "Attribute Value as String: " << sf.ToString( tDoseGridScaling ) << std::endl;

        // Let's check the name again:
        std::pair<std::string, std::string> pss
            = sf.ToStringPair( tDoseGridScaling );
        std::cout << "Attribute Name Checked: " << pss.first << std::endl;
        std::cout << "Attribute Value (string): " << pss.second << std::endl;

        //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

        // Let's assume for a moment we knew the tag number:
        Attribute<0x3004,0x000e> at;
        gdcm_assert( at.GetTag() == tDoseGridScaling );
        at.SetFromDataSet( ds );
        // For the sake of long term maintenance, we will not write
        // that this particular attribute is stored as a double. What if
        // a user made a mistake. It is much safer to rely on GDCM internal
        // mechanism to deduce the VR::DS type (represented as a ieee double)
        Attribute<0x3004,0x000e>::ArrayType v = at.GetValue();
    }
}

```

```

    std::cout << "DoseGridScaling=" << v << std::endl;
}

return 0;
}

```

14.91 ReadExplicitLengthSQIVR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmByteValue.h"
#include "gdcmDataSet.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmReader.h"
#include "gdcmSequenceOfItems.h"

using namespace gdcm;

int main(int argc, char *argv[])
{
    if ( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader r;
    r.SetFileName( filename );
    r.Read();

    //gdcm::PrivateTag pt(0xe1,0x42,"ELSCINT1");
    //gdcm::Tag pt(0x88,0x200);
    gdcm::Tag pt(0x8,0x1140);
    DataSet &ds = r.GetFile().GetDataSet();
    const DataElement &de = ds.GetDataElement( pt );

    std::cout << de << std::endl;
    const ByteValue *bv = de.GetByteValue();
    SmartPointer<SequenceOfItems> sqi = new SequenceOfItems;
    sqi->SetLength( bv->GetLength() );
    std::stringstream ss;
    ss.str( std::string( bv->GetPointer(), bv->GetLength() ) );
    sqi->Read<ImplicitDataElement,SwapperNoOp>( ss );

    std::cout << *sqi << std::endl;

    return 0;
}

```

14.92 ReadGEMSSDO.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```



```

===== */
#include "gdcmlReader.h"
#include "gdcmlDataElement.h"
#include "gdcmlPrivateTag.h"

#include <iostream>
#include <string>

using namespace gdcml;

struct SDOElement
{
    typedef std::vector<std::string>::size_type SizeType;
    const char *GetData(SizeType index) const {
        return Data[index].c_str();
    }
    SizeType GetNumberOfData() const {
        return Data.size();
    }
    void SetData(SizeType index, const char *data) {
        Data[index] = data;
    }
    const char *GetDataFormat() const {
        return DataFormat.c_str();
    }
    void SetDataFormat(const char *dataformat, SizeType num) {
        DataFormat = dataformat;
        Data.resize( num );
    }
    void Print( std::ostream &os ) const {
        os << DataFormat << " " << std::endl;
        std::vector<std::string>::const_iterator it = Data.begin();
        size_t s = 0;
        for( ; it != Data.end(); ++it )
        {
            os << " (" << s++ << " ) " << *it << std::endl;
        }
    }
private:
    std::string DataFormat;
    std::vector<std::string> Data;
};

class SDOHeader
{
public:
    typedef std::vector<SDOElement> SDOElements;
    typedef SDOElements::size_type SizeType;
    SizeType GetNumberOfSDOElements() const {
        return InternalSDODDataSet.size();
    }
    void AddSDOElement(SDOElement const &sdoelement) {
        InternalSDODDataSet.push_back( sdoelement );
    }
    const SDOElement &GetSDOElement(SizeType index) const {
        return InternalSDODDataSet[index];
    }
    const SDOElement &GetSDOElementByName(const char *) const {
        return InternalSDODDataSet[0];
    }
    void LoadFromAttributes(std::string const &s1, std::string const &s2)
    {
        std::string tok;
        std::string tok2;
        std::stringstream strstr(s1);
        std::stringstream strstr2(s2);

        SDOElement element;
        // Do format
        size_t count = 0;
        while ( std::getline ( strstr2, tok, '\\' ) )
        {
            //std::cout << tok << " ";
            std::getline ( strstr2, tok2, '\\' );
            //std::cout << tok2 << std::endl;
            count += atoi( tok2.c_str() );
            element.SetDataFormat( tok.c_str(), atoi( tok2.c_str() ) );
            for( size_t t = 0; t < element.GetNumberOfData(); ++t )
            {
                std::getline ( strstr, tok, '\\' );
                element.SetData(t, tok.c_str() );
            }
        }
    }
};

```

```

    }
    AddSDOElement( element );
}
//while ( std::getline ( strstr, tok, '^' ) )
// while ( std::getline ( strstr, tok, '\\' ) )
// {
//     std::cout << tok << std::endl;
//     count++;
// }
// std::cout << "Count: " << count << std::endl;
// count = 0;

// std::cout << "Count: " << count << std::endl;

}
void Print( std::ostream &os ) const {
    SDOElements::const_iterator it = InternalSDODataset.begin();
    for( ; it != InternalSDODataset.end(); ++it )
    {
        it->Print ( os );
    }
}
private:
    SDOElements InternalSDODataset;
};

bool sdo_decode( DataElement const &stringdata, DataElement const &stringdataformat )
{
    const char *sd = stringdata.GetByteValue()->GetPointer();
    const size_t len_sd = stringdata.GetByteValue()->GetLength();

    std::string s1 = std::string( sd, len_sd );

    const char *sdf = stringdataformat.GetByteValue()->GetPointer();
    const size_t len_sdf = stringdataformat.GetByteValue()->GetLength();

    std::string s2 = std::string( sdf, len_sdf );

    // std::cout << s1 << std::endl;
    // std::cout << s2 << std::endl;

    SDOHeader header;
    header.LoadFromAttributes( s1, s2 );

    header.Print( std::cout );

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    File &file = reader.GetFile();
    DataSet &ds = file.GetDataSet();

    // StringData (0033,xx1F) 3 "GEMS_GENIE_1" List of SDO parameters stored as
    // list of strings
    const PrivateTag tstringdata(0x33,0x1f,"GEMS_GENIE_1");
    // StringDataFormat (0033,xx23) 3 "GEMS_GENIE_1" Format of string parameters;
    // contains information about name and number of strings in list
    const PrivateTag tstringdataformat(0x33,0x23,"GEMS_GENIE_1");

    if( !ds.FindDataElement( tstringdata ) ) return 1;
    const DataElement& stringdata = ds.GetDataElement( tstringdata );
    if( !ds.FindDataElement( tstringdataformat ) ) return 1;

```

```

const DataElement& stringdataformat = ds.GetDataElement( tstringdataformat );

sdo_decode( stringdata, stringdataformat );

return 0;
}

```

14.93 ReadMultiTimesException.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
// The intention of this sample program is to provoke bad_alloc exceptions in gdcm code

#include "gdcmImageReader.h"

int main(int argc, char* argv[])
{
    // We pre-allocate some memory (about 1Gb) to help the issue to show up earlier
    char *dummyBuffer = new char[1024*1024*1100]; (void)dummyBuffer;
    // Check the number of parameters given
    if (argc < 3)
    {
        std::cerr << "Usage: " << argv[0] << " Filename numberOfTries" << std::endl;
        return 1;
    }

    std::cout << "We are going to read the file: " << argv[1] << " " << argv[2] << " times" << std::endl;
    // We hold the pointers in an array to avoid the memory to be released
    // We read the input file n-times
    for (int i = 0; i < atoi(argv[2]); ++i)
    {
        gdcm::ImageReader reader;
        std::cout << "Reading try: " << i << std::endl;
        // Read files
        reader.SetFileName(argv[1]);
        try
        {
            reader.Read();
            gdcm::Image & img = reader.GetImage();
            unsigned long len = img.GetBufferLength();
            char *buffer = new char[ len ];
            img.GetBuffer( buffer ); // do NOT de-allocate buffer !
        }
        catch (std::bad_alloc &ba)
        {
            (void)ba;
            std::cerr << "BAD ALLOC Exception caught!" << std::endl;
        }
        catch (...)
        {
            std::cerr << "Exception caught!" << std::endl;
        }
    }

    return 0;
}

```

14.94 ReadUTF8QtDir.cxx

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcml.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

===== */
/*
 * GDCM API expect a const char * as input for SetFileName
 * In order to use this API from Qt, here is a simple test that
 * shows how to do it in a portable manner:
 *
 * http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
 */

#include "gdcmlReader.h"
#include "gdcmlDirectory.h"

#include <QDir>
#include <QString>
#include <QCoreApplication>

#include <string>
#include <fstream>

#include <stdio.h> // fopen

static int TestBothFuncs(const char *info , const char *ba_str)
{
    int res = 0;
    FILE *f = fopen( ba_str, "r" );
    if( f )
    {
        std::cout << info << " fopen: " << ba_str << std::endl;
        fclose(f);
        ++res;
    }
    gdcml::Reader reader;
    std::ifstream is( ba_str, std::ios::binary );
    if( is.is_open() )
    {
        std::cout << info << " is_open: " << ba_str << std::endl;
        ++res;
    }
    reader.SetStream( is );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetStream/CanRead:" << ba_str << std::endl;
        ++res;
    }
    is.close();
    reader.SetFileName( ba_str );
    if( reader.CanRead() == true )
    {
        std::cout << info << " SetFileName/CanRead:" << ba_str << std::endl;
        ++res;
    }
    return 4 - res;
}

static int scanFolder(const char dirname[])
{
    int res = 0;
    gdcml::Directory dir;
    unsigned int nfiles = dir.Load( dirname, true );
    const gdcml::Directory::FileNamesType &filenames = dir.GetFileNames();

    for( unsigned int i = 0; i < nfiles; ++i )
    {
        const char *ba_str = filenames[i].c_str();
        res += TestBothFuncs("GDCM",ba_str);
    }
    return res;
}

static int scanFolderQt(QDir const &dir, QStringList& files)
{

```

```

int res = 0;
QFileInfoList children = dir.entryInfoList(QDir::AllEntries|QDir::NoDotAndDotDot);
for ( int i=0; i<children.count(); i++ ) {
    QFileInfo file = children.at(i);
    if ( file.isDir() == true ) {
        res += scanFolderQt(QDir(file.absoluteFilePath()), files);
        continue;
    }
    // Convert back from the internal representation to 8bits
    // toLocal8Bit() returns by copy. Need to store explicitly the QByteArray
    QByteArray str = file.absoluteFilePath().toLocal8Bit();
    const char *ba_str1 = str.constData();
    res += TestBothFuncs("QString", ba_str1);
}
return res;
}

int main(int argc, char *argv[])
{
    // very important:
    QCoreApplication qCoreApp( argc , argv );
    if( argc < 2 )
    {
        std::cerr << argv[0] << " dir " << std::endl;
        return 1;
    }

    int res = 0;
    const char *dirname = argv[1];
    res += scanFolder( dirname );

    QDir dir( QString::fromLocal8Bit(dirname) );
    QStringList files;
    res += scanFolderQt( dir, files);

    if( res )
        std::cerr << "Problem with UTF-8" << std::endl;
    else
        std::cerr << "Success with UTF-8" << std::endl;

    return res;
}

```

14.95 SimpleScanner.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Simple example to show how to use Scanner API.
* It exposes the three different cases:
* - DICOM Attribute is present and has a value
* - DICOM Attribute is present and has no value
* - DICOM Attribute is not present at all
* It also shows the purpose of the function 'IsKey' to detect whether or
* not the file has been read by the gdcms::Scanner. Technically most of the time
* if a file is not a 'Key' this is because it is not a DICOM file. You need to use
* gdcms::System::FileExists to decide whether or not the file actually exist on the disk.
*
* It was tested on this particular image:
* ./SimpleScanner gdcmsData/012345.002.050.dcm
*/

#include "gdcmsStrictScanner.h"
#include "gdcmsSimpleSubjectWatcher.h"

```

```

#include "gdcmFileNameEvent.h"

class MyFileWatcher : public gdcm::SimpleSubjectWatcher
{
public:
    MyFileWatcher(gdcm::Subject *s, const char *comment = ""):
        gdcm::SimpleSubjectWatcher(s,comment){}
    void ShowFileName(gdcm::Subject *, const gdcm::Event &evt) override
    {
        const gdcm::FileNameEvent &pe = dynamic_cast<const gdcm::FileNameEvent&>(evt);
        const char *fn = pe.GetFileName();
        std::cout << "FileName: " << fn << " FileSize: " << gdcm::System::FileSize( fn ) << std::endl;
    }
};

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char filename_invalid[] = "this is a file that may not exist on this disk.dcm";

    gdcm::SmartPointer<gdcm::StrictScanner> sp = new gdcm::StrictScanner;
    gdcm::StrictScanner &s = *sp;
    //gdcm::SimpleSubjectWatcher w(&s, "TestFileName" );
    MyFileWatcher w(&s, "TestFileName" );

    const gdcm::Tag tag_array[] = {
        gdcm::Tag(0x8,0x50),
        gdcm::Tag(0x8,0x51),
        gdcm::Tag(0x8,0x60),
        gdcm::Tag(0x8,0x80),
    };
    s.AddTag( tag_array[0] );
    s.AddTag( tag_array[1] );
    s.AddTag( tag_array[2] );
    s.AddTag( tag_array[3] );

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );
    filenames.push_back( filename_invalid );

    if( !s.Scan( filenames ) )
    {
        return 1;
    }

    //s.Print( std::cout );

    for(gdcm::Directory::FileNamesType::const_iterator it = filenames.begin();
        it != filenames.end(); ++it )
    {
        if( s.IsKey( it->c_str() ) )
        {
            std::cout << "INFO:" << it->c_str() << " is a proper Key for the Scanner (this is a DICOM file)" << std::endl;
        }
        else
        {
            std::cout << "INFO:" << it->c_str() << " is not a proper Key for the Scanner (this is either not a DICOM file or file does not exist)"
                << std::endl;
        }
    }

    gdcm::StrictScanner::TagToValue const &ttv = s.GetMapping(filename);

    const gdcm::Tag *ptag = tag_array;
    for( ; ptag != tag_array + 3; ++ptag )
    {
        gdcm::StrictScanner::TagToValue::const_iterator it = ttv.find( *ptag );
        if( it != ttv.end() )
        {
            std::cout << *ptag << " was properly found in this file" << std::endl;
            // it contains a pair of value. the first one is the actual tag, so the following is always true:
            // *ptag == it->first
            // The second part is the actual value (stored as RAW strings). You will have to reinterpret this string
            // if VR for *ptag is not VR::VRASCII !
            const char *value = it->second;

```

```

    if( *value )
    {
        std::cout << " It has the value: " << value << std::endl;
    }
    else
    {
        std::cout << " It has no value (empty)" << std::endl;
    }
}
else
{
    std::cout << "Sorry " << *ptag << " could not be found in this file" << std::endl;
}
}

return 0;
}

```

14.96 SortImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
*/
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
    gdcm::Attribute<0x0018,0x1060> at1; // Trigger Time
    gdcm::Attribute<0x0020,0x0032> at11; // Image Position (Patient)
    at1.Set( ds1 );
    at11.Set( ds1 );
    //gdcm::Attribute<0x0020,0x0013> at2;
    gdcm::Attribute<0x0018,0x1060> at2;
    gdcm::Attribute<0x0020,0x0032> at22;
    at2.Set( ds2 );
    at22.Set( ds2 );
    if( at11 == at22 )
    {
        return at1 < at2;
    }
    return at11 < at22;
}

bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0018,0x1060> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0018,0x1060> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x0032> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0032> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

```

```

}

// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0052> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

int main(int argc, char *argv[])
{
    if (argc < 2 ) return 1;
    const char *dirname = argv[1];
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname );

    dir.Print( std::cout );

    gdcm::Sorter sorter;
    sorter.SetSortFunction( mysort );
    sorter.Sort( dir.GetFileNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdcm::Sorter sorter2;
    sorter2.SetSortFunction( mysort_part1 );
    sorter2.StableSort( dir.GetFileNames() );
    sorter2.SetSortFunction( mysort_part2 );
    sorter2.StableSort( sorter2.GetFileNames() ); // IMPORTANT
    sorter2.SetSortFunction( mysort_dummy );
    sorter2.StableSort( sorter2.GetFileNames() ); // IMPORTANT

    std::cout << "Sorter2:" << std::endl;
    sorter2.Print( std::cout );

    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFileNames() );

    //s.Print( std::cout );

    // Count how many different IPP there are:
    const gdcm::Scanner::ValueType &values = s.GetValues();
    size_t nvalues = values.size();
    std::cout << "There are " << nvalues << " different type of values" << std::endl;

    //std::cout << "nfiles=" << nfiles << std::endl;
    if( nfiles % nvalues != 0 )
    {
        std::cerr << "Impossible: this is a not a proper series" << std::endl;
        return 1;
    }
    std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D volumes" << std::endl;

    return 0;
}

```

14.97 StreamImageReaderTest.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```



```

===== */
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmStreamImageReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmSystem.h"
#include "gdcmFilename.h"
#include "gdcmByteSwap.h"
#include "gdcmTrace.h"
#include "gdcmTesting.h"
#include "gdcmImageHelper.h"
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmMediaStorage.h"
#include "gdcmRAWCodec.h"
#include "gdcmJPEGCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

bool StreamImageRead(gdcm::StreamImageWriter & theStreamWriter,
    const char* filename, const char* outfilename, int resolution)
{
    gdcm::StreamImageReader reader;

    reader.SetFileName( filename );

    if (!reader.ReadImageInformation())
    {
        std::cerr << "unable to read image information" << std::endl;
        return 1; //unable to read tags as expected.
    }
    //let's be tricky; each image will be read in portions, first the top half, then the bottom
    //that way, we can test how the stream handles fragmentation of the data
    //we could also loop this to get various different size combinations, but I'm not sure
    //that's useful, yet.
    std::vector<unsigned int> extent =
        gdcm::ImageHelper::GetDimensionsValue(reader.GetFile());
    // std::cout << extent[0];
    //at this point, these values aren't used, but may be in the future
    //unsigned short xmin = 0;
    //unsigned short xmax = extent[0];
    //unsigned short ymin = 0;
    //unsigned short ymax = extent[1];
    //unsigned short zmin = 0;
    //unsigned short zmax = extent[2];

    std::cout << "\n Row: " << extent[0] << "\n Col : " << extent[1] << "\n Resolution : " << extent[2] << std::endl;

    int a = 1;
    for (int i=1; i<=(extent[2]-resolution);++i)
        a = a*2;

    reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a, resolution-1, resolution);

    unsigned long len = reader.DefineProperBufferLength();
    char* finalBuffer = new char[len];
    memset(finalBuffer, 0, sizeof(char)*len);

    if (reader.CanReadImage())
    {
        bool result = reader.Read(finalBuffer, len);
        if( !result )
        {
            std::cout << "res2 failure:" << filename << std::endl;
            delete [] finalBuffer;
            return 1;
        }
        else
        {
            std::cout << "Able to read";
        }
    }
    else
    {
        std::cerr << "Not able to put in buffer" << std::endl;
    }
}
/*
    //now, read in smaller buffer extents

```

```

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
len = reader.DefineProperBufferLength();

char* buffer = new char[len];
bool res2 = reader.Read(buffer, len);
if( !res2 ){
    std::cerr << "res2 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(finalBuffer, buffer, len);

//now read the next half of the image
ymin = ymax;
ymax = extent[1];

reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

//std::cerr << "Success to read image from file: " << filename << std::endl;
unsigned long len2 = reader.DefineProperBufferLength();

char* buffer2 = new char[len2];
bool res3 = reader.Read(buffer2, len2);
if( !res3 ){
    std::cerr << "res3 failure:" << filename << std::endl;
    return 1;
}
//copy the result into finalBuffer
memcpy(&(finalBuffer[len]), buffer2, len2);

delete [] buffer;
delete [] buffer2;
*/

gdcm::Writer w;
gdcm::File &file = w.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

gdcm::UIDGenerator uid;
gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
de.SetVR( gdcm::VR::UI );
const char *u = uid.Generate();
de.SetByteValue( u, strlen(u) );
ds.Insert( de );

gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
de1.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
de1.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( de1 );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0010> row = {extent[0]/a};//
ds.Insert( row.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0011> col = {extent[1]/a};//
ds.Insert( col.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {1};//
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*

```

```

    ds1.Remove( gdcmm::Tag(0x0028,0x0008) );

    gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
    ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
    theStreamWriter.SetFile(file);

    if (!theStreamWriter.WriteImageInformation())
    {
        std::cerr << "unable to write image information" << std::endl;
        return 1; //the CanWrite function should prevent getting here, else,
        //that's a test failure
    }
    std::vector<unsigned int> extent1 = gdcmm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent1[0];
    unsigned short ymax = extent1[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = 1;

    std::cout << "\n Row: " << extent1[0] << "\n Col : " << extent1[1] << "\n Resolution : " << extent1[2] << std::endl;

    if (xmax == 0 || ymax == 0)
    {
        std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
        return 0;
    }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially. So, store how far you got in the buffer with each iteration.

    for (z = 0; z < zmax; ++z){
        for (y = 0; y < ymax; y += ychunk){
            nexty = y + ychunk;
            if (nexty > ymax) nexty = ymax;
            theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
            unsigned long len = theStreamWriter.DefineProperBufferLength();
            std::cout << "\n" << len;
            char* finalBuffer1 = new char[len];
            memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
            std::cout << "\nable to write";

            if (!theStreamWriter.Write(finalBuffer1, len)){
                std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z= " << z << std::endl;
                delete [] finalBuffer1;
                delete [] finalBuffer;
                return 1;
            }
            delete [] finalBuffer1;
            prevLen += len;
        }
    }
    delete [] finalBuffer;
    std::cout << "all is set";

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm Resolution" << std::endl;
        return 1;
    }

    const char *filename = argv[1];
    const char *outfilename = argv[2];
    char *res = argv[3];

    int resolution = atoi(res);

    gdcmm::StreamImageWriter theStreamWriter;

    std::ofstream of;
    of.open( outfile, std::ios::out | std::ios::binary );

```

```

theStreamWriter.SetStream(of);

// else
// First of get rid of warning/debug message
gdcm::Trace::DebugOn();
gdcm::Trace::WarningOn();

if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;

uint16_t firstTag1 = 0xffff;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize1];
memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
//memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
gdcm_assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize1);
of.flush();
gdcm_assert( of );

return 0;
}

```

14.98 TemplateEmptyImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmFileStreamer.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmImageRegionReader.h"
#include "gdcmImageHelper.h"
#include "gdcmWriter.h"
#include "gdcmImageWriter.h"
#include "gdcmTagKeywords.h"
#include "gdcmUIDGenerator.h"

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char * filename = argv[1];
    gdcm::ImageRegionReader irr;
    irr.SetFileName( filename );
    const bool b3 = irr.ReadInformation();
    std::cout << b3 << std::endl;
    gdcm::Image & img = irr.GetImage();
    std::cout << img << std::endl;
    // const gdcm::Region & r = irr.GetRegion();
    // std::cout << r << std::endl;
    gdcm::ImageWriter w;
    gdcm::File & file = w.GetFile();
    gdcm::DataSet & ds = file.GetDataSet();

    gdcm::UIDGenerator uid;
    namespace kwd = gdcm::Keywords;
    kwd::FrameOfReferenceUID frameref;
    frameref.SetValue( uid.Generate() );
    // ContentDate
    char date[22];
    const size_t datelen = 8;
    int res = gdcm::System::GetCurrentDateTime(date);

```

```

(void)res;
kwd::ContentDate contentdate;
// Do not copy the whole cstring:
contentdate.SetValue( gdcmm::DComp( date, datelen ) );
ds.Insert( contentdate.GetAsDataElement() );
// ContentTime
const size_t timelen = 6 + 1 + 6; // time + milliseconds
kwd::ContentTime contenttime;
// Do not copy the whole cstring:
contenttime.SetValue( gdcmm::TComp( date+datelen, timelen ) );
ds.Insert( contenttime.GetAsDataElement() );
gdcmm::MediaStorage ms0 = w.ComputeTargetMediaStorage();
std::cout << ms0 << std::endl;
kwd::SeriesNumber seriesnumber = { 1 };
kwd::InstanceNumber instancenum = { 1 };
kwd::StudyID studyid = { "St1" };
kwd::PatientID patientid = { "P1" };
kwd::SOPClassUID sopclassuid;
kwd::PositionReferenceIndicator pri;
//kwd::Laterality lat;
//kwd::BodyPartExamined bodypartex = { "HEAD" };
kwd::BodyPartExamined bodypartex = { "ANKLE" };
kwd::PatientOrientation pator;
kwd::BurnedInAnnotation bia = { "NO" };
kwd::ConversionType convtype = { "SYN" };
kwd::PresentationLUTShape plutshape = { "IDENTITY" }; // MONOCHROME2
// gdcmm will pick the Word in case Byte class is not compatible:
gdcmm::MediaStorage ms = gdcmm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage;
sopclassuid.SetValue( ms.GetString() );
ds.Insert( instancenum.GetAsDataElement() );
ds.Insert( sopclassuid.GetAsDataElement() );
ds.Insert( seriesnumber.GetAsDataElement() );
ds.Insert( patientid.GetAsDataElement() );
ds.Insert( studyid.GetAsDataElement() );
ds.Insert( frameref.GetAsDataElement() );
ds.Insert( pri.GetAsDataElement() );
//ds.Insert( lat.GetAsDataElement() );
ds.Insert( bodypartex.GetAsDataElement() );
ds.Insert( pator.GetAsDataElement() );
ds.Insert( bia.GetAsDataElement() );
ds.Insert( convtype.GetAsDataElement() );
ds.Insert( plutshape.GetAsDataElement() );
// gdcmm::MediaStorage ms1 = w.ComputeTargetMediaStorage();
// std::cout << ms1 << std::endl;
std::cout << ds << std::endl;
gdcmm::PixelFormat & pf = img.GetPixelFormat();
pf.SetPixelRepresentation(0); // always overwrite
img.SetSlope(1);
img.SetIntercept(0);
w.SetImage( img );
w.SetFileName( "TemplateImage.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

14.99 TraverseModules.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

```

```

*/

#include "gdcDefs.h"
#include "gdcGlobal.h"
#include "gdcIODs.h"
#include "gdcIOD.h"
#include "gdcMacros.h"
#include "gdcIODEntry.h"
#include "gdcModules.h"
#include "gdcModule.h"
#include "gdcAnonymizer.h"
#include "gdcDicts.h"

int main(int , char *[])
{
    using namespace gdc;
    static Global &g = Global::GetInstance();

    if( !g.LoadResourcesFiles() )
    {
        return 1;
    }

    static const Defs &defs = g.GetDefs();
    static const Modules &modules = defs.GetModules();
    static const IODs &iods = defs.GetIODs();
    static const Macros &macros = defs.GetMacros();
    static const Dicts &dicts = g.GetDicts();

    std::vector<Tag> tags = gdc::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
    for( std::vector<Tag>::const_iterator tit = tags.begin(); tit != tags.end(); ++tit )
    {
        const Tag &tag = *tit;
        const DictEntry &dictentry = dicts.GetDictEntry(tag);
        std::cout << "Processing Attribute: " << tag << " " << dictentry << std::endl;

        IODs::IODMapTypeConstIterator it = iods.Begin();
        for( ; it != iods.End(); ++it )
        {
            const IODs::IODName &name = it->first;
            const IOD &iod = it->second;

            const size_t niods = iod.GetNumberOfIODs();
            // Iterate over each iod entry in order:
            for(unsigned int idx = 0; idx < niods; ++idx)
            {
                const IODEntry &iodentry = iod.GetIODEntry(idx);
                const char *ref = iodentry.GetRef();
                //Usage::UsageType ut = iodentry.GetUsageType();

                const Module &module = modules.GetModule( ref );
                if( module.FindModuleEntryInMacros(macros, tag ) )
                {
                    const ModuleEntry &module_entry = module.GetModuleEntryInMacros(macros,tag);
                    Type type = module_entry.GetType();
                    std::cout << "IOD Name: " << name << std::endl;
                    std::cout << "Type: " << type << std::endl;
                }
            }
        }
    }

    return 0;
}

```

14.100 VolumeSorter.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even

```

the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
/*
*/
#include "gdcmSorter.h"
#include "gdcmIPPSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
#include "gdcmTesting.h"

bool mysort1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x000d> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x000d> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    gdcm::Attribute<0x0020,0x000e> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x000e> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort3(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    // This is a floating point number is the comparison ok ?
    gdcm::Attribute<0x0020,0x0037> at1;
    at1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0037> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort4(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
    // Do the IPP sorting here
    gdcm::Attribute<0x0020,0x0032> ipp1;
    gdcm::Attribute<0x0020,0x0037> iop1;
    ipp1.Set( ds1 );
    iop1.Set( ds1 );
    gdcm::Attribute<0x0020,0x0032> ipp2;
    gdcm::Attribute<0x0020,0x0037> iop2;
    ipp2.Set( ds2 );
    iop2.Set( ds2 );
    if( iop1 != iop2 )
    {
        return false;
    }

    // else
    double normal[3];
    normal[0] = iop1[1]*iop1[5] - iop1[2]*iop1[4];
    normal[1] = iop1[2]*iop1[3] - iop1[0]*iop1[5];
    normal[2] = iop1[0]*iop1[4] - iop1[1]*iop1[3];
    double dist1 = 0;
    for (int i = 0; i < 3; ++i) dist1 += normal[i]*ipp1[i];
    double dist2 = 0;
    for (int i = 0; i < 3; ++i) dist2 += normal[i]*ipp2[i];

    std::cout << dist1 << ", " << dist2 << std::endl;
    return dist1 < dist2;
}

int main(int argc, char *argv[])
{
    const char *extradataroot = gdcm::Testing::GetDataExtraRoot();
    std::string dirl;
    if( argc < 2 )
    {
        if( !extradataroot )

```

```

    {
        return 1;
    }
    dir1 = extradataroot;
    dir1 += "/gdcmsampleData/ForSeriesTesting/VariousIncidences/ST1";
}
else
{
    dir1 = argv[1];
}

gdcms::Directory d;
d.Load( dir1, true ); // recursive !
const gdcms::Directory::FileNamesType &l1 = d.GetFilesNames();
const size_t nfiles = l1.size();
std::cout << nfiles << std::endl;

//if( nfiles != 280 )
// {
//     return 1;
// }

//d.Print( std::cout );

gdcms::Scanner s0;
const gdcms::Tag t1(0x0020,0x000d); // Study Instance UID
const gdcms::Tag t2(0x0020,0x000e); // Series Instance UID
//const gdcms::Tag t3(0x0010,0x0010); // Patient's Name
s0.AddTag( t1 );
s0.AddTag( t2 );
//s0.AddTag( t3 );
//s0.AddTag( t4 );
//s0.AddTag( t5 );
//s0.AddTag( t6 );
bool b = s0.Scan( d.GetFilesNames() );
if( !b )
{
    std::cerr << "Scanner failed" << std::endl;
    return 1;
}

//s0.Print( std::cout );

// Only get the DICOM files:
gdcms::Directory::FileNamesType l2 = s0.GetKeys();
const size_t nfiles2 = l2.size();
std::cout << nfiles2 << std::endl;

if ( nfiles2 > nfiles )
{
    return 1;
}

gdcms::Sorter sorter;
sorter.SetSortFunction( mysort1 );
sorter.StableSort( l2 );

sorter.SetSortFunction( mysort2 );
sorter.StableSort( sorter.GetFilesNames() );

sorter.SetSortFunction( mysort3 );
sorter.StableSort( sorter.GetFilesNames() );

sorter.SetSortFunction( mysort4 );
sorter.StableSort( sorter.GetFilesNames() );

//sorter.Print( std::cout );

// Let's try to check our result:
// assume that IPP is precise enough so that we can test floating point equality:
size_t nvalues = 0;
{
    gdcms::Scanner s;
    s.AddTag( gdcms::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcms::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFilesNames() );

    //s.Print( std::cout );

    const gdcms::Scanner::ValuesType &values = s.GetValues();

```



```

nvalues = values.size();
std::cout << "There are " << nvalues << " different type of values" << std::endl;
gdcm::assert( nfiles2 % nvalues == 0 );
std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D volumes" << std::endl;
}

gdcm::Directory::FileNamesType sorted_files = sorter.GetFileNames();

// Which means we can take nvalues files at a time and execute gdcm::IPPSorter on it:
gdcm::IPPSorter ippsorter;
gdcm::Directory::FileNamesType sub( sorted_files.begin(), sorted_files.begin() + nvalues);
std::cout << sub.size() << std::endl;
std::cout << sub[0] << std::endl;
std::cout << sub[nvalues-1] << std::endl;
ippsorter.SetComputeZSpacing( false );
if( !ippsorter.Sort( sub ) )
{
    std::cerr << "Could not sort" << std::endl;
    return 1;
}

std::cout << "IPPSorter:" << std::endl;
ippsorter.Print( std::cout );

return 0;
}

```

14.101 csa2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* I do not know what the format is, just guessing from info found on the net:
*
* http://atonal.ucdavis.edu/matlab/fmri/spm5/spm_dicom_convert.m
*
* This example is an attempt at understanding the format used by SIEMENS
* their "SIEMENS CSA NON-IMAGE" DICOM file (1.3.12.2.1107.5.9.1)
*
* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // gdcmDataExtra/gdcmNonImageData/exCSA_Non-Image_Storage.dcm
    // PHANTOM.MR.CARDIO_COEUR_S_QUENCE_DE_REP_RAGE.9.257.2008.03.20.14.53.25.578125.43151705.IMA
    const char *filename = argv[1];

    gdcm::Reader reader; // Do not use ImageReader
    reader.SetFileName( filename );
    if( !reader.Read() )

```

```

{
    std::cerr << "Failed to read: " << filename << std::endl;
    return 1;
}

gdcmm::CSAHeader csa;
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

const gdcmm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
//std::cout << t1 << std::endl;
//const gdcmm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

if( ds.FindElement( t1 ) )
{
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
    csa.Print( std::cout );
}
int dims[2] = {};
if( csa.FindCSAElementByName( "Columns" ) )
{
    const gdcmm::CSAElement &csael = csa.GetCSAElementByName( "Columns" );
    std::cout << csael << std::endl;
    //const gdcmm::ByteValue *bv = csael.GetByteValue();
    gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1> el;
    el.Set( csael.GetValue() );
    dims[0] = el.GetValue();
    std::cout << "Columns:" << el.GetValue() << std::endl;
}

if( csa.FindCSAElementByName( "Rows" ) )
{
    const gdcmm::CSAElement &csael2 = csa.GetCSAElementByName( "Rows" );
    std::cout << csael2 << std::endl;
    gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1> el2;
    el2.Set( csael2.GetValue() );
    dims[1] = el2.GetValue();
    std::cout << "Rows:" << el2.GetValue() << std::endl;
}

double spacing[2] = { 1., 1. };
bool spacingfound = false;
if( csa.FindCSAElementByName( "PixelSpacing" ) )
{
    const gdcmm::CSAElement &csael3 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !csael3.IsEmpty() )
    {
        std::cout << csael3 << std::endl;
        gdcmm::Element<gdcmm::VR::DS, gdcmm::VM::VM2> el3;
        el3.Set( csael3.GetValue() );
        spacing[0] = el3.GetValue(0);
        spacing[1] = el3.GetValue(1);
        std::cout << "PixelSpacing:" << el3.GetValue() << "," << el3.GetValue(1) << std::endl;
        spacingfound = true;
    }
}

if( !spacingfound )
{
    std::cerr << "Problem with PixelSpacing" << std::endl;
    //return 1;
}
if( !dims[0] || !dims[1] )
{
    std::cerr << "Problem with dims" << std::endl;
    return 1;
}

gdcmm::ImageWriter writer;

gdcmm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 2 ); // good default
image.SetDimension(0, dims[0]);
image.SetDimension(1, dims[1]);
image.SetSpacing(0, spacing[0]);
image.SetSpacing(1, spacing[1]);
gdcmm::PixelFormat pixeltypes = gdcmm::PixelFormat::INT16; // bytepix = spm_type('int16','bits')/8;

//unsigned long l = image.GetBufferLength();
//const int p = 1 / (dims[0] * dims[1]);

//image.SetNumberOfDimensions( 3 );

```

```

//image.SetDimension(2, p / pixeltype.GetPixelSize() );

gdcmm::PhotometricInterpretation pi;
pi = gdcmm::PhotometricInterpretation::MONOCHROME2;
//pixeltype.SetSamplesPerPixel( );
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );
//image.SetIntercept( inputimage.GetIntercept() );
//image.SetSlope( inputimage.GetSlope() );

//gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe1,0x1010) );
//pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
gdcmm::PrivateTag csananimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
const gdcmm::DataElement &pixeldata = ds.GetDataElement( csananimaget );
image.SetDataElement( pixeldata );

std::string outfilename = "outcsa.dcm";
//writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

14.102 iU22tomultisc.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcmmReader.h"
#include "gdcmmImageWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // IM_001
    const char *filename = argv[1];

    gdcmm::Reader reader; // Do not use ImageReader
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }

    // * The data is simply 8-bit unsigned in the obvious x/y/z order
    // * 200D,300B contains the data
    // * 200D,3001 contains the no. of voxels (416,412,256 in this case)
    // * 200D,3003 contains the voxel sizes (0.156184527398215 /
    // 0.1223749613981957 / 0.328479990704639 in this case)

    const gdcmm::File &file = reader.GetFile();
    const gdcmm::DataSet &ds = file.GetDataSet();
    const gdcmm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );

```

```

const gdcm::DataElement &rawdatus = ds.GetDataElement( trawdatus );

const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
const gdcm::DataElement &tcolsrowsframes = ds.GetDataElement( tcolsrowsframes );
// const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging DD 036" );
// this is just a duplicate previous tag.
const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips US Imaging DD 036" );
const gdcm::DataElement &tvoxelspacing = ds.GetDataElement( tvoxelspacing );

gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> dims; // Use DS to interpret value stored in LO
dims.SetFromDataElement( colsrowsframes );

gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> spacing;
spacing.SetFromDataElement( voxelspacing );

gdcm::ImageWriter writer;

gdcm::Image &image = writer.GetImage();
image.SetNumberOfDimensions( 3 ); // good default
image.SetDimension(0, (unsigned int)dims[0] );
image.SetDimension(1, (unsigned int)dims[1] );
image.SetDimension(2, (unsigned int)dims[2] );
image.SetSpacing(0, spacing[0] );
image.SetSpacing(1, spacing[1] );
image.SetSpacing(2, spacing[2] );
gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8;

gdcm::PhotometricInterpretation pi;
pi = gdcm::PhotometricInterpretation::MONOCHROME2;
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );

image.SetDataElement( rawdatus );

std::string outfilename = "outiu22.dcm";

gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage );
// gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
writer.GetFile().GetDataSet().Replace( de );

writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

14.103 pmsct_rgb1.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* This example shows how to rewrite a ELSCINT1/PMSCT_RGB1 compressed
* image so that it is readable by most 3rd party software (DICOM does
* not specify this particular encoding).
* This is required for the sake of interoperability with any standard
* conforming DICOM system.
*

```

```

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcmm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Jean-Pierre Roux for providing the sample datasets
*/
#include "gdcmmReader.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmAttribute.h"
#include "gdcmmImageWriter.h"

void delta_decode(const unsigned char *data_in, size_t data_size,
                 std::vector<unsigned char> &new_stream, unsigned short pc, size_t w, size_t h)
{
    const size_t plane_size = h * w;
    const size_t outputlen = 3 * plane_size;
    new_stream.resize( outputlen );

    gdcmm_assert( data_size != outputlen );
    if( data_size == outputlen )
    {
        return;
    }
    typedef unsigned char byte;
    enum {
        COLORMODE = 0x81,
        ESCMODE = 0x82,
        REPEATMODE = 0x83
    };

    const byte* src = (const byte*)data_in;
    byte* dest = (byte*)new_stream.data();
    union { byte gray; byte rgb[3]; } pixel;
    pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
    // always start in grayscale mode
    bool graymode = true;
    size_t dx = 1;
    size_t dy = 3;
    // algorithm works with both planar configuration
    // It does produce surprising greenish background color for planar
    // configuration is 0, while the nested Icon SQ display a nice black
    // background
    if (pc)
    {
        dx = plane_size;
        dy = 1;
    }
    size_t ps = plane_size;

    // The following is highly unoptimized as we have nested if statement in a while loop
    // we need to switch from one algorithm to the other (RGB <-> GRAY)
    while (ps)
    {
        // next byte:
        byte b = *src++;
        gdcmm_assert( src < data_in + data_size );
        // mode selection:
        switch ( b )
        {
            {
                case ESCMODE:
                    // Used to treat a byte 81/82/83 as a normal byte
                    if (graymode)
                    {
                        pixel.gray += *src++;
                        dest[0*dx] = pixel.gray;
                        dest[1*dx] = pixel.gray;
                        dest[2*dx] = pixel.gray;
                    }
                    else
                    {
                        pixel.rgb[0] += *src++;
                        pixel.rgb[1] += *src++;
                        pixel.rgb[2] += *src++;
                        dest[0*dx] = pixel.rgb[0];
                        dest[1*dx] = pixel.rgb[1];
                    }
                }
            }
        }
    }
}

```

```

        dest[2*dx] = pixel.rgb[2];
    }
    dest += dy;
    ps--;
    break;
case REPEATMODE:
    // repeat mode (RLE)
    b = *src++;
    ps -= b;
    if (graymode)
    {
        while (b-- > 0)
        {
            dest[0*dx] = pixel.gray;
            dest[1*dx] = pixel.gray;
            dest[2*dx] = pixel.gray;
            dest += dy;
        }
    }
    else
    {
        while (b-- > 0)
        {
            dest[0*dx] = pixel.rgb[0];
            dest[1*dx] = pixel.rgb[1];
            dest[2*dx] = pixel.rgb[2];
            dest += dy;
        }
    }
    break;
case COLORMODE:
    // We are switching from one mode to the other. The stream contains an intermixed
    // compression of RGB codec and GRAY codec. Each one not knowing of the other
    // reset old value to 0.
    if (graymode)
    {
        graymode = false;
        pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
    }
    else
    {
        graymode = true;
        pixel.gray = 0;
    }
    break;
default:
    // This is identical to ESCMODE, it would be nicer to use fall-through
    if (graymode)
    {
        pixel.gray += b;
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
    }
    else
    {
        pixel.rgb[0] += b;
        pixel.rgb[1] += *src++;
        pixel.rgb[2] += *src++;
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
    }
    dest += dy;
    ps--;
    break;
} // end switch
} // end while
}

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
}

```

```

const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

// (07a1,1011) CS [PMSCT_RGB1] # 10,1 Tamar Compression Type
const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
if( !ds.FindDataElement( tcompressiontype ) ) return 1;
const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
if ( compressiontype.IsEmpty() ) return 1;
const gdcm::ByteValue * bv = compressiontype.GetByteValue();
std::string comprle = "PMSCT_RLE1";
std::string comprgb = "PMSCT_RGB1";
bool isrle = false;
bool isrgb = false;
if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
{
    isrle = true;
    return 1;
}
if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
{
    isrgb = true;
}
if( !isrgb && !isrle ) return 1;

const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"ELSCINT1");
if( !ds.FindDataElement( tcompressedpixeldata ) ) return 1;
const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tcompressedpixeldata );
if ( compressionpixeldata.IsEmpty() ) return 1;
const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

gdcm::Attribute<0x0028,0x0006> at0;
at0.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

std::vector<unsigned char> buffer;
delta_decode((const unsigned char*)bv2->GetPointer(), bv2->GetLength(), buffer,
    at0.GetValue(), at1.GetValue(), at2.GetValue() );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcm::VR::OW );
pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)buffer.size() );
// TODO we should check that decompress byte buffer match the expected size (row*col*...)

// Add the pixel data element
reader.GetFile().GetDataSet().Replace( pixeldata );

reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
gdcm::Writer writer;
writer.SetFile( reader.GetFile() );

// Cleanup stuff:
// remove the compressed pixel data:
// FIXME: should I remove more private tags ? all of them ?
// oh well this is just an example
// use gdcm::Anonymizer::RemovePrivateTags if needed...
writer.GetFile().GetDataSet().Remove( compressionpixeldata.GetTag() );
std::string outfilename;
if (argc > 2)
    outfilename = argv[2];
else
    outfilename = "outrgb.dcm";
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "Failed to write" << std::endl;
    return 1;
}

std::cout << "success !" << std::endl;

return 0;
}

```

14.104 rle2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSC_T_RLE1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcms-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Mauro Maiorca for bringing to our attention on this new ELSCINT1
 * compression algorithm : PMSC_T_RLE1 (different from the 'LOSSLESS RICE')
 * See post at:
 * http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
 *
 * Thanks to Jesus Spinola, for more datasets,
 * http://www.itk.org/pipermail/insight-users/2008-April/025571.html
 *
 * And last but not least, a very big thank to Ivo van Poorten, without
 * whom we would still be looking at this compressed byte stream as if
 * it was RLE compressed.
 */
#include "gdcmsReader.h"
#include "gdcmsPrivateTag.h"
#include "gdcmsAttribute.h"
#include "gdcmsImageWriter.h"

/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ... */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned short> &output)
{
    // RLE pass
    std::vector<char> temp;
    for(size_t i = 0; i < length; ++i)
    {
        if( inbuffer[i] == (char)0xa5 )
        {
            //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
            //gdcms_assert( (unsigned char)inbuffer[i+1] != 255 );
            int repeat = (unsigned char)inbuffer[i+1] + 1;
            char value = inbuffer[i+2];
            while(repeat)
            {
                temp.push_back( value );
                --repeat;
            }
            i+=2;
        }
        else
        {
            temp.push_back( inbuffer[i] );
        }
    }

    // Delta encoding pass
    unsigned short delta = 0;
    for(size_t i = 0; i < temp.size(); ++i)
    {
        if( temp[i] == 0x5a )

```



```

    {
        unsigned char v1 = (unsigned char)temp[i+1];
        unsigned char v2 = (unsigned char)temp[i+2];
        unsigned short value = (unsigned short)(v2 * 256 + v1);
        output.push_back( value );
        delta = value;
        i+=2;
    }
    else
    {
        unsigned short value = (unsigned short)((signed char)temp[i] + delta);
        output.push_back( value );
        delta = value;
    }
    //gdcmm_assert( output[output.size()-1] == ref[output.size()-1] );
}

if ( output.size() % 2 )
{
    output.resize( output.size() - 1 );
}
std::cout << "length " << output.size() * 2 << std::endl;
}

int main(int argc, char *argv [])
{
    if( argc < 2 )
    {
        std::cerr << "input.dcm [output.dcm]" << std::endl;
        std::cerr << "will default to 'out.rle.dcm' unless output.dcm is specified."
        << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    // (07a1,1011) CS [PMSCT_RLE1] # 10,1 Tamar Compression Type
    const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
    if( !ds.FindDataElement( tcompressiontype ) ) return 1;
    const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
    if ( compressiontype.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = compressiontype.GetByteValue();
    std::string comprle = "PMSCT_RLE1";
    std::string comprgb = "PMSCT_RGB1";
    bool isrle = false;
    bool isrgb = false;
    if( strcmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
        isrle = true;
    }
    if( strcmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
    {
        isrgb = true;
        std::cerr << "See: pmsct_rgb1.cxx instead" << std::endl;
        return 1;
    }
    if( !isrgb && !isrle ) return 1;

    // check if compressed pixel data reside in private or standard tag
    const gdcm::PrivateTag tprivatepixeldata(0x07a1,0x100a,"ELSCINT1");
    const gdcm::Tag tstandardpixeldata(0x7fe0, 0x0010);
    gdcm::Tag tpixeldata;
    if(ds.FindDataElement(tpprivatepixeldata)) tpixeldata = tprivatepixeldata;
    else if(ds.FindDataElement(tstandardpixeldata)) tpixeldata = tstandardpixeldata;
    if(!ds.FindDataElement(tpixeldata)) return 1;

    const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tpixeldata);
    if ( compressionpixeldata.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

    gdcm::Attribute<0x0028,0x0010> at1;
    at1.SetFromDataSet( ds );
    gdcm::Attribute<0x0028,0x0011> at2;

```

```

at2.SetFromDataSet( ds );

gdcmm::DataElement pixeldata;
// if standard voxel data element does not exist, create it
if( !reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
    pixeldata = gdcmm::DataElement( tpixeldata, 0, gdcmm::VR::OW );
}
else{
    pixeldata = reader.GetFile().GetDataSet().GetDataElement( tpixeldata );
}

pixeldata.SetVR( gdcmm::VR::OW );
gdcmm::VL bv2l = bv2->GetLength();
gdcmm::VL at1l = at1.GetValue() * at2.GetValue() * 2; /* sizeof(unsigned short) == 2 */
// Handle special case that is not compressed:
if( bv2l == at1l )
{
    pixeldata.SetByteValue( bv2->GetPointer(), bv2->GetLength() );
}
else
{
    std::vector<unsigned short> buffer;
    delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);
    pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)(buffer.size() * sizeof( unsigned short )) );
}
// TODO we should check that decompress byte buffer match the expected size (row*col*...)

// Add the pixel data element
if( reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
    reader.GetFile().GetDataSet().Replace( pixeldata );
}
else
{
    reader.GetFile().GetDataSet().ReplaceEmpty( pixeldata );
}

reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian);
gdcmm::Writer writer;
writer.SetFile( reader.GetFile() );

// Cleanup stuff:
// This makes the code equivalent to Philips workstation IntelliSpace Portal
if( writer.GetFile().GetDataSet().FindDataElement( tcompressiontype ) )
{
    writer.GetFile().GetDataSet().Remove( gdcmm::Tag(0x07a1,0x1011) );
}
if( writer.GetFile().GetDataSet().FindDataElement( tprivatepixeldata ) )
{
    writer.GetFile().GetDataSet().Remove( gdcmm::Tag(0x07a1,0x100a) );
}

std::string outfilename;
if (argc > 2)
    outfilename = argv[2];
else
    outfilename = "outrlc.dcm";
writer.SetFileName( outfilename.c_str() );
if( !writer.Write() )
{
    std::cerr << "Failed to write" << std::endl;
    return 1;
}

std::cout << "success !" << std::endl;

return 0;
}

```

14.105 uid_unique.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdc.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```
=====*/
#include "gdcUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
    gdc::UIDGenerator uid;
    //const char myroot[] = "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
    const char myroot[] = "9876543210.9876543210.9876543210";
    uid.SetRoot( myroot );
    std::set<std::string> uids;
    uint64_t wrap = 0;
    uint64_t c = 0;
    while(true)
    {
        const char *unique = uid.Generate();
        //std::cout << unique << std::endl;
        if( c % 10000 == 0 )
        {
            std::cout << "wrap=" << wrap << ",c=" << c << std::endl;
        }
        ++c;
        if( c == 0 )
        {
            wrap++;
        }
        if ( uids.count(unique) == 1 )
        {
            std::cerr << "Failed with: " << unique << std::endl;
            return 1;
        }
        uids.insert( unique );
    }
}
```

14.106 DecompressImage.java

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdc.jar javac ../../gdc/Examples/Java/DecompressImage.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdc.jar: java DecompressImage gdcData/012345.002.050.dcm out.dcm
 */
import gdc.*;

public class DecompressImage
{
```

```

public static void main(String[] args) throws Exception
{
    String file1 = args[0];
    String file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
    {
        throw new Exception("Could not read: " + file1 );
    }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    change.SetInput( reader.GetImage() );
    if( !change.Change() )
    {
        throw new Exception("Could not change: " + file1 );
    }

    Image out = change.GetOutput();
    System.out.println( out.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( out );
    ret = writer.Write();
    if( !ret )
    {
        throw new Exception("Could not write: " + file2 );
    }
}
}

```

14.107 DecompressPixmap.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../gdcm/Examples/Java/DecompressPixmap.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressPixmap gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressPixmap
{
    public static void main(String[] args) throws Exception
    {
        String file1 = args[0];
        String file2 = args[1];
        PixmapReader reader = new PixmapReader();
        reader.SetFileName( file1 );
        boolean ret = reader.Read();
        if( !ret )

```

```

    {
        throw new Exception("Could not read: " + file1 );
    }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    PixmapToPixmapFilter filter = (PixmapToPixmapFilter)change;
    filter.SetInput( reader.GetPixmap() );
    if( !change.Change() )
    {
        throw new Exception("Could not change: " + file1 );
    }

    // The following does not work in Java/swig 2.0.7
    //Pixmap p = ((PixmapToPixmapFilter)change).GetOutput();
    Pixmap p = change.GetOutputAsPixmap(); // be explicit
    //System.out.println( p.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( p );
    ret = writer.Write();
    if( !ret )
    {
        throw new Exception("Could not write: " + file2 );
    }
    }
}

```

14.108 ExtractImageRegion.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java ExtractImageRegion input.dcm
 */
import gdcm.*;
import java.io.FileOutputStream;

public class ExtractImageRegion
{
    public static void main(String[] args) throws Exception
    {
        String filename = args[0];

        // instantiate the reader:
        ImageRegionReader reader = new ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return;
        // Get file infos
        File f = reader.GetFile();

        // get some info about image
    }
}

```

```

UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
int pixelsize = pf.GetPixelSize();

// buffer to get the pixels
long buffer_length = dims.get(0) * dims.get(1) * pixelsize;
byte[] buffer = new byte[ (int)buffer_length ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (int z = 0; z < dims.get(2); z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims.get(0) - 1, 0, dims.get(1) - 1, z, z);
    //System.Console.WriteLine( box.toString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // long buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, buffer_length))
    {
        FileOutputStream fos = new FileOutputStream("/tmp/frame.raw");
        fos.write(buffer);
        fos.close();
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}
}
}

```

14.109 FileAnonymize.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import gdcm.*;

public class FileAnonymize
{
    public static class MyWatcher extends SimpleSubjectWatcher
    {
        public MyWatcher(Subject s) { super(s,"Override String"); }
        protected void ShowProgress(Subject caller, Event evt)
        {
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.out.println( "This is my progress: " + pe.GetProgress() );
        }
    }

    public static void main(String[] args) throws Exception
    {
        String input = args[0];
        String output = args[1];

        FileAnonymizer fa = new FileAnonymizer();
        fa.SetInputFileName( input );
    }
}

```

```

fa.SetOutputFileName( output );

// Empty Operations
// It will create elements, since those tags are non-registered public elements (2011):
fa.Empty( new Tag(0x0008,0x1313) );
fa.Empty( new Tag(0x0008,0x1317) );
// Remove Operations
// The following Tag are actually carefully chosen, since they refer to SQ:
fa.Remove( new Tag(0x0008,0x2112) );
fa.Remove( new Tag(0x0008,0x9215) );
// Replace Operations
// do not call replace operation on SQ attribute !
fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

if( !fa.Write() )
{
    System.out.println( "Could not write" );
    return;
}

System.out.println( "success" );
}
}

```

14.110 HelloSimple.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
* Compilation:
* $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
*
* Usage:
* $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java HelloSimple gdcmData/012345.002.050.dcm
*/
import gdcm.*;

public class HelloSimple
{
    public static void main(String[] args) throws Exception
    {
        String filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + filename );
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        System.out.println( ds.toString() );

        System.out.println("Success reading: " + filename );
    }
}

```

14.111 ReadFiles.java

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcml.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
import gdcml.*;
import java.io.File;

public class ReadFiles
{
    static int i = 0;
    public static void process(String path)
    {
        //String path = file.getPath();
        assert PosixEmulation.FileExists(path) : "Problem converting to 8bits";

        System.out.println("Reading: " + path );
        System.out.println("File: " + i++);
        Reader r = new Reader();
        try
        {
            r.SetFileName( path );
            TagSetType skip = new TagSetType();
            skip.insert( new Tag(0x7fe0,0x10) );
            boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
            //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
        }
        finally
        {
            r.delete(); // will properly call C++ destructor and close file descriptor
        }
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public static void waiting (int n)
    {
        long t0, t1;
        t0 = System.currentTimeMillis();
        do
        {
            t1 = System.currentTimeMillis();
        }
        while ((t1 - t0) < (n * 1000));
    }

    public static void main(String[] args) throws Exception
    {
        String directory = args[0];

        Directory gdir = new Directory();
        long n = gdir.Load( directory, true );
        System.out.println( gdir.toString() );
        FilenamesType files = gdir.GetFilenames();
        for( long i = 0; i < n; ++i )
        {
            String path = files.get( (int)i );
            process( path );
        }
    }
}

```



```

    }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
    {
        File dir = new File(directory);
        visitAllFiles(dir);
    }
}

```

14.112 ScanDirectory.java

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

import gdcm.*;
import gdcm.Reader;
import gdcm.LookupTable;
import java.io.File;
import java.io.*;
import java.awt.image.*;
import javax.imageio.ImageIO;

public class ScanDirectory
{
    public static class MyWatcher extends SimpleSubjectWatcher
    {
        public MyWatcher(Subject s) { super(s,"Override String"); }
        protected void ShowProgress(Subject caller, Event evt)
        {
            ProgressEvent pe = ProgressEvent.Cast(evt);
            System.out.println( "This is my progress: " + pe.GetProgress() );
        }
    }

    public static byte[] GetAsByte(Bitmap input)
    {
        long len = input.GetBufferLength();
        byte[] buffer = new byte[ (int)len ];
        PhotometricInterpretation pi = input.GetPhotometricInterpretation();
        if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
        {
            ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
            icpi.SetInput( input );
            icpi.SetPhotometricInterpretation(
                new PhotometricInterpretation(
                    PhotometricInterpretation.PIType.MONOCHROME2 ) );
            if( icpi.Change() )
            {
                Bitmap output = icpi.GetOutput();
                output.GetArray( buffer );
            }
            return buffer;
        }
        else
        {
            input.GetArray( buffer );
            return buffer;
        }
    }

    public static short[] GetAsShort(Bitmap input)
    {
        long len = input.GetBufferLength(); // length in bytes

```

```

short[] buffer = new short[ (int)len / 2 ];
PhotometricInterpretation pi = input.GetPhotometricInterpretation();
if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
{
    ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
    icpi.SetInput( input );
    icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
            PhotometricInterpretation.PIType.MONOCHROME2 ) );
    if( icpi.Change() )
    {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
    }
    return buffer;
}
else
{
    input.GetArray( buffer );
    return buffer;
}
}

public static boolean WritePNG(Bitmap input, String outfilename )
{
    int imageType = BufferedImage.TYPE_CUSTOM;
    PixelFormat pf = input.GetPixelFormat();
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    // We need to handle both public and private icon
    // It could well be that we are getting an RGB Icon or 16 bits Icon:
    ColorModel colorModel = null;
    if( pf.GetSamplesPerPixel() == 1 )
    {
        if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1
            || pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME2 )
        {
            if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
            {
                imageType = BufferedImage.TYPE_BYTE_GRAY;
            }
            else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT12 )
            {
                imageType = BufferedImage.TYPE_USHORT_GRAY;
            }
            else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT16 )
            {
                imageType = BufferedImage.TYPE_USHORT_GRAY;
            }
        }
        else if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
        {
            LookupTable lut = input.GetLUT();
            long rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
            byte[] rbuf = new byte[ (int)rl ];
            long rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
            assert rl == rl2;
            long gl = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
            byte[] gbuf = new byte[ (int)gl ];
            long gl2 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
            assert gl == gl2;
            long bl = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
            byte[] bbuf = new byte[ (int)bl ];
            long bl2 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
            assert bl == bl2;
            colorModel = new IndexColorModel(8, (int)rl, rbuf, gbuf, bbuf);
            // For code below
            imageType = BufferedImage.TYPE_BYTE_GRAY;
        }
    }
    else if( pf.GetSamplesPerPixel() == 3 )
    {
        if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
            // FIXME should be TYPE_3BYTE_RGB
            imageType = BufferedImage.TYPE_3BYTE_BGR;
        }
    }
}

//System.out.println( "pf: " + pf.toString() );
//System.out.println( "pi: " + pi.toString() );
long width = input.GetDimension(0);
long height = input.GetDimension(0);
BufferedImage bi;

```

```

if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
{
    bi = new BufferedImage(colorModel,
        colorModel.createCompatibleWritableRaster((int)width, (int)height),
        false, null);
}
else
{
    bi = new BufferedImage((int)width,(int)height,imageType);
}
WritableRaster wr = bi.getRaster();
//System.out.println( "imageType: " + imageType );
if( imageType == BufferedImage.TYPE_BYTE_GRAY
    || imageType == BufferedImage.TYPE_3BYTE_BGR )
{
    byte[] buffer = GetAsByte( input );
    wr.setDataElements (0, 0, (int)width, (int)height, buffer);
}
else if( imageType == BufferedImage.TYPE_USHORT_GRAY )
{
    short[] buffer = GetAsShort( input );
    wr.setDataElements (0, 0, (int)width, (int)height, buffer);
}

File outputfile = new File( outfilename );
try {
    ImageIO.write(bi, "png", outputfile);
} catch (IOException e) {
    return false;
}
return true;
}

public static void main(String[] args) throws Exception
{
    String directory = args[0];

    Directory d = new Directory();
    long nfiles = d.Load( directory, true );
    if(nfiles == 0)
    {
        throw new Exception("No files found");
    }
    // System.out.println( "Files:\n" + d.toString() );
    FilenamesType fns = d.GetFilenames();

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    MyWatcher watcher = new MyWatcher(s);
    Tag[] tagarray = {
        new Tag(0x0010, 0x0010), // PatientName
        new Tag(0x0010, 0x0020), // PatientID
        new Tag(0x0010, 0x0030), // PatientBirthDate
        new Tag(0x0010, 0x0040), // PatientSex
        new Tag(0x0010, 0x1010), // PatientAge
        new Tag(0x0020, 0x000d), // StudyInstanceUID
        new Tag(0x0020, 0x0010), // StudyID
        new Tag(0x0008, 0x0020), // StudyDate
        new Tag(0x0008, 0x1030), // StudyDescription
        new Tag(0x0020, 0x000e), // SeriesInstanceUID
        new Tag(0x0020, 0x0011), // SeriesNumber
        new Tag(0x0008, 0x0021), // SeriesDate
        new Tag(0x0008, 0x103e), // SeriesDescription
        new Tag(0x0008, 0x0090), // ReferringPhysicianName
        new Tag(0x0008, 0x0060), // Modality
        new Tag(0x0054, 0x0400), // ImageID ?? Should be Instance number ??
        new Tag(0x0008, 0x0018), // SOPInstanceUID
        new Tag(0x0008, 0x0032), // AcquisitionTime
        new Tag(0x0008, 0x0033), // ContentTime
        new Tag(0x0020, 0x0013), // InstanceNumber
        new Tag(0x0020, 0x1041), // SliceLocation
        new Tag(0x0018, 0x0050), // SliceThickness ?? Eg. Enhanced MR Image Storage
        new Tag(0x0008, 0x0080), // InstitutionName
        new Tag(0x0028, 0x1050), // WindowCenter
        new Tag(0x0028, 0x1051), // WindowWidth
    };
    for( Tag t : tagarray ) {
        //System.out.println( "Tag: " + t.toString() );
        s.AddTag( t );
    }
}

```

```

    }
    boolean b = s.Scan( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    String fn0 = fns.get(0);
    TagToValue mappings = s.GetMapping( fn0 );
    System.out.println( "mappings size: " + mappings.size() );
    for( Tag tag : tagarray ) {
        if( mappings.has_key( tag ) ) {
            String val = mappings.get( tag );
            System.out.println( "tag/val: " + tag + "->" + val );
        }
    }

    for( long idx = 0; idx < fns.size(); ++idx )
    {
        Reader r = new Reader();
        String fn = fns.get( (int)idx );
        String outfn = fn + ".png";
        r.SetFileName( fn );
        TagSetType tst = new TagSetType();
        tst.insert( new Tag(0x7fe0,0x10) );
        b = r.ReadUpToTag( new Tag(0x88,0x200), tst );
        UIntArrayType dims = ImageHelper.GetDimensionsValue( r.GetFile() );
        if( b )
        {
            IconImageFilter iif = new IconImageFilter();
            System.out.println( "Processing: " + fn );

            iif.SetFile( r.GetFile() );
            b = iif.Extract();
            if( b )
            {
                Bitmap icon = iif.GetIconImage(0);
                WritePNG(icon, outfn);
            }
        }
        else
        {
            ImageReader ir = new ImageReader();
            ir.SetFileName( fn );
            if( ir.Read() )
            {
                Image img = ir.GetImage();
                StringFilter sf = new StringFilter();
                sf.SetFile( r.GetFile() );
                String strval = sf.ToString( new Tag(0x0028,0x0120) );
                IconImageGenerator iig = new IconImageGenerator();
                iig.SetPixmap( img );
                iig.AutoPixelMinMax( true );
                try {
                    double val = Double.parseDouble( strval );
                    iig.SetOutsideValuePixel( val );
                }
                catch ( NumberFormatException e ) {
                }
                iig.ConvertRGBToPaletteColor( false );
                long idims[] = { 128, 128 };
                iig.SetOutputDimensions( idims );
                iig.Generate();
                Bitmap icon = iig.GetIconImage();
                WritePNG(icon, outfn);
            }
        }
    }
}

System.out.println( "Scan:\n" + s.toString() );

System.out.println( "success" );
}
}

```

14.113 SimplePrint.java

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/SimplePrint.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java SimplePrint gdcmData/012345.002.050.dcm
 */
import gdcmm.*;

public class SimplePrint
{
    public static void RecurseDataSet(File f, DataSet ds, String indent)
    {
        JavaDataSet cds = new JavaDataSet(ds);
        while(!cds.IsAtEnd())
        {
            DataElement de = cds.GetCurrent();
            // Compute VR from the toplevel file, and the currently processed dataset:
            VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

            if( vr.Compatible( new VR(VR.VRType.SQ) ) )
            {
                long uvl = de.GetVL().GetValueLength(); // Test cast is ok
                System.out.println( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
                //SequenceOfItems sq = de.GetSequenceOfItems();
                // GetValueAsSQ handle more cases than GetSequenceOfItems
                SmartPtrSQ sq = de.GetValueAsSQ();
                long n = sq.GetNumberOfItems();
                for( long i = 1; i <= n; i++) // item starts at 1, not 0
                {
                    Item item = sq.GetItem( i );
                    DataSet nested = item.GetNestedDataSet();
                    RecurseDataSet( f, nested, indent + " " );
                }
            }
            else
            {
                System.out.println( indent + de.toString() );
            }
            cds.Next();
        }
    }

    public static void main(String[] args) throws Exception
    {
        String filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + filename );
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        RecurseDataSet( f, ds, "" );
    }
}

```

14.114 AddPrivateAttribute.py

```

00001
00014
00015 """

```

```

00016 Usage:
00017
00018 python AddPrivateAttribute.py input.dcm output.dcm
00019
00020
00021 """
00022
00023 import sys
00024 import gdcm
00025
00026 if __name__ == "__main__":
00027
00028     file1 = sys.argv[1]
00029     file2 = sys.argv[2]
00030
00031     r = gdcm.Reader()
00032     r.SetFileName( file1 )
00033     if not r.Read():
00034         sys.exit(1)
00035
00036     f = r.GetFile()
00037     ds = f.GetDataSet()
00038
00039     # Create a dataelement
00040     de = gdcm.DataElement(gdcm.Tag(0x0051, 0x1011))
00041     de.SetByteStringValue("p2")
00042     de.SetVR(gdcm.VR(gdcm.VR.SH))
00043
00044     ds.Insert(de)
00045
00046     w = gdcm.Writer()
00047     w.SetFile( f )
00048     w.SetFileName( file2 )
00049     if not w.Write():
00050         sys.exit(1)

```

14.115 ConvertMPL.py

```

00001
00014
00015 """
00016 display a DICOM image with matplotlib via numpy
00017
00018 Caveats:
00019 - Does not support UINT12/INT12
00020
00021 Usage:
00022
00023 python ConvertNumpy.py "IM000000"
00024
00025 Thanks:
00026 plotting example - Ray Schumacher 2009
00027 """
00028
00029 import gdcm
00030 import numpy
00031 from pylab import *
00032
00033
00034 def get_gdcm_to_numpy_typemap():
00035     """Returns the GDCM Pixel Format to numpy array type mapping."""
00036     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
00037                 gdcm.PixelFormat.INT8 :numpy.uint8,
00038                 gdcm.PixelFormat.UINT16 :numpy.uint16,
00039                 gdcm.PixelFormat.INT16 :numpy.int16,
00040                 gdcm.PixelFormat.UINT32 :numpy.uint32,
00041                 gdcm.PixelFormat.INT32 :numpy.int32,
00042                 gdcm.PixelFormat.FLOAT32:numpy.float32,
00043                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
00044     return _gdcm_np
00045
00046 def get_numpy_array_type(gdcm_pixel_format):
00047     """Returns a numpy array typecode given a GDCM Pixel Format."""
00048     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00049
00050 def gdcm_to_numpy(image):
00051     """Converts a GDCM image to a numpy array.

```

```

00052     """
00053     pf = image.GetPixelFormat().GetScalarType()
00054     print 'pf', pf
00055     print image.GetPixelFormat().GetScalarTypeAsString()
00056     assert pf in get_gdcm_to_numpy_typemap().keys(), \
00057         "Unsupported array type %s"%pf
00058     d = image.GetDimension(0), image.GetDimension(1)
00059     print 'Image Size: %d x %d' % (d[0], d[1])
00060     dtype = get_numpy_array_type(pf)
00061     gdcm_array = image.GetBuffer()
00062
00063     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
00064
00068     result.shape = d
00069     return result
00070
00071 if __name__ == "__main__":
00072     import sys
00073     r = gdcm.ImageReader()
00074     filename = sys.argv[1]
00075     r.SetFileName( filename )
00076     if not r.Read(): sys.exit(1)
00077     numpy_array = gdcm_to_numpy( r.GetImage() )
00078
00079     subplot(111)# one plot, on left
00080     title(filename)
00081
00082     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
00083
00084     subplots_adjust(bottom=0.1, right=0.8, top=0.9)
00085     cax = axes([0.85, 0.1, 0.075, 0.8])
00086     colorbar(cax=cax)
00087     title('values')
00088     get_current_fig_manager().window.title('plot')
00089     show()

```

14.116 ConvertNumpy.py

```

00001
00014
00015 """
00016 This module add support for converting a gdcm.Image to a numpy array.
00017
00018 Caveats:
00019 - Does not support UINT12/INT12
00020
00021 Removed:
00022 - float16 is defined in GDCM API but no implementation exist for it ...
00023 """
00024
00025 import gdcm
00026 import numpy
00027
00028 def get_gdcm_to_numpy_typemap():
00029     """Returns the GDCM Pixel Format to numpy array type mapping."""
00030     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.uint8,
00031                 gdcm.PixelFormat.INT8  :numpy.int8,
00032                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
00033                 #gdcm.PixelFormat.INT12  :numpy.int12,
00034                 gdcm.PixelFormat.UINT16 :numpy.uint16,
00035                 gdcm.PixelFormat.INT16  :numpy.int16,
00036                 gdcm.PixelFormat.UINT32 :numpy.uint32,
00037                 gdcm.PixelFormat.INT32  :numpy.int32,
00038                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
00039                 gdcm.PixelFormat.FLOAT32:numpy.float32,
00040                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
00041     return _gdcm_np
00042
00043 def get_numpy_array_type(gdcm_pixel_format):
00044     """Returns a numpy array typecode given a GDCM Pixel Format."""
00045     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00046
00047 def gdcm_to_numpy(image):
00048     """Converts a GDCM image to a numpy array.
00049     """
00050     pf = image.GetPixelFormat()
00051

```

```

00052     assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
00053         "Unsupported array type %s"%pf
00054
00055     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
00056     if image.GetNumberOfDimensions() == 3:
00057         shape = shape[0] * image.GetDimension(2), shape[1]
00058
00059     dtype = get_numpy_array_type(pf.GetScalarType())
00060     gdcm_array = image.GetBuffer()
00061     result = numpy.frombuffer(gdcm_array, dtype=dtype)
00062     result.shape = shape
00063     return result
00064
00065 if __name__ == "__main__":
00066     import sys
00067     r = gdcm.ImageReader()
00068     filename = sys.argv[1]
00069     r.SetFileName( filename )
00070     if not r.Read():
00071         sys.exit(1)
00072
00073     numpy_array = gdcm_to_numpy( r.GetImage() )
00074     print numpy_array

```

14.117 ConvertPIL.py

```

00001
00014
00015 """
00016 save a DICOM image with PIL via numpy
00017
00018 Caveats:
00019 - Does not support UINT12/INT12
00020
00021 Usage:
00022
00023 python ConvertNumpy.py "IM000000"
00024
00025 Thanks:
00026 plotting example - Ray Schumacher 2009
00027 """
00028
00029 import gdcm
00030 import numpy
00031 from PIL import Image, ImageOps
00032
00033
00034 def get_gdcm_to_numpy_typemap():
00035     """Returns the GDCM Pixel Format to numpy array type mapping."""
00036     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
00037                 gdcm.PixelFormat.INT8 :numpy.uint8,
00038                 gdcm.PixelFormat.UINT16 :numpy.uint16,
00039                 gdcm.PixelFormat.INT16 :numpy.int16,
00040                 gdcm.PixelFormat.UINT32 :numpy.uint32,
00041                 gdcm.PixelFormat.INT32 :numpy.int32,
00042                 gdcm.PixelFormat.FLOAT32:numpy.float32,
00043                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
00044     return _gdcm_np
00045
00046 def get_numpy_array_type(gdcm_pixel_format):
00047     """Returns a numpy array typecode given a GDCM Pixel Format."""
00048     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
00049
00050 def gdcm_to_numpy(image):
00051     """Converts a GDCM image to a numpy array.
00052     """
00053     pf = image.GetPixelFormat().GetScalarType()
00054     print 'pf', pf
00055     print image.GetPixelFormat().GetScalarTypeAsString()
00056     assert pf in get_gdcm_to_numpy_typemap().keys(), \
00057         "Unsupported array type %s"%pf
00058     d = image.GetDimension(0), image.GetDimension(1)
00059     print 'Image Size: %d x %d' % (d[0], d[1])
00060     dtype = get_numpy_array_type(pf)
00061     gdcm_array = image.GetBuffer()
00062     result = numpy.frombuffer(gdcm_array, dtype=dtype)
00063     maxV = float(result[result.argmax()])

```



```

00064
00067     result = numpy.log(result+50)
00068     maxV = float(result[result.argmax()])
00069     result = result*(2.**8/maxV)
00070     result.shape = d
00071     return result
00072
00073 if __name__ == "__main__":
00074     import sys
00075     r = gdcm.ImageReader()
00076     filename = sys.argv[1]
00077     r.SetFileName( filename )
00078     if not r.Read(): sys.exit(1)
00079     numpy_array = gdcm_to_numpy( r.GetImage() )
00080
00082     pilImage = Image.frombuffer('L',
00083                                numpy_array.shape,
00084                                numpy_array.astype(numpy.uint8),
00085                                'raw','L',0,1)
00086
00087     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
00088     pilImage.save(sys.argv[1]+'jpg')

```

14.118 CreateRAWStorage.py

```

00001
00014
00015 """
00016 <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP Class" part="PS 3.4" retired="false"/>
00017 """
00018
00019 import gdcm
00020 import sys,os
00021
00022 if __name__ == "__main__":
00023     r = gdcm.Reader()
00024     # Will require Testing...
00025     dataroot = gdcm.Testing.GetDataRoot()
00026     filename = os.path.join( dataroot, '012345.002.050.dcm' )
00027     r.SetFileName( filename )
00028     r.Read()
00029     f = r.GetFile()
00030     ds = f.GetDataSet()
00031
00032     uid = "1.2.840.10008.5.1.4.1.1.66"
00033     # f = gdcm.File()
00034     # ds = f.GetDataSet()
00035     de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
00036     de.SetByteStringValue( uid )
00037     vr = gdcm.VR( gdcm.VR.UI )
00038     de.SetVR( vr )
00039     ds.Replace( de )
00040
00041     ano = gdcm.Anonymizer()
00042     ano.SetFile( r.GetFile() )
00043     ano.RemovePrivateTags()
00044     ano.RemoveGroupLength()
00045     taglist = [
00046         gdcm.Tag(0x0008,0x0008),
00047         gdcm.Tag(0x0008,0x0022),
00048         gdcm.Tag(0x0008,0x0032),
00049         gdcm.Tag(0x0008,0x2111),
00050         gdcm.Tag(0x0008,0x1150),
00051         gdcm.Tag(0x0008,0x1155),
00052         gdcm.Tag(0x0008,0x0100),
00053         gdcm.Tag(0x0008,0x0102),
00054         gdcm.Tag(0x0008,0x0104),
00055         gdcm.Tag(0x0040,0xa170),
00056         gdcm.Tag(0x0008,0x2112),
00057         gdcm.Tag(0x0008,0x0100),
00058         gdcm.Tag(0x0008,0x0102),
00059         gdcm.Tag(0x0008,0x0104),
00060         gdcm.Tag(0x0008,0x9215),
00061         gdcm.Tag(0x0018,0x0010),
00062         gdcm.Tag(0x0018,0x0022),
00063         gdcm.Tag(0x0018,0x0050),
00064         gdcm.Tag(0x0018,0x0060),

```

```

00065  gdc.Tag(0x0018,0x0088),
00066  gdc.Tag(0x0018,0x0090),
00067  gdc.Tag(0x0018,0x1040),
00068  gdc.Tag(0x0018,0x1100),
00069  gdc.Tag(0x0018,0x1110),
00070  gdc.Tag(0x0018,0x1111),
00071  gdc.Tag(0x0018,0x1120),
00072  gdc.Tag(0x0018,0x1130),
00073  gdc.Tag(0x0018,0x1150),
00074  gdc.Tag(0x0018,0x1151),
00075  gdc.Tag(0x0018,0x1152),
00076  gdc.Tag(0x0018,0x1160),
00077  gdc.Tag(0x0018,0x1190),
00078  gdc.Tag(0x0018,0x1210),
00079  gdc.Tag(0x0020,0x0012),
00080  gdc.Tag(0x0020,0x0032),
00081  gdc.Tag(0x0020,0x0037),
00082  gdc.Tag(0x0020,0x1041),
00083  gdc.Tag(0x0020,0x4000),
00084  gdc.Tag(0x0028,0x0002),
00085  gdc.Tag(0x0028,0x0004),
00086  gdc.Tag(0x0028,0x0010),
00087  gdc.Tag(0x0028,0x0011),
00088  gdc.Tag(0x0028,0x0030),
00089  gdc.Tag(0x0028,0x0100),
00090  gdc.Tag(0x0028,0x0101),
00091  gdc.Tag(0x0028,0x0102),
00092  gdc.Tag(0x0028,0x0103),
00093  gdc.Tag(0x0028,0x1052),
00094  gdc.Tag(0x0028,0x1053),
00095  gdc.Tag(0x0028,0x2110),
00096  gdc.Tag(0x0028,0x2112),
00097  gdc.Tag(0x7fe0,0x0010),
00098  gdc.Tag(0x0018,0x0020),
00099  gdc.Tag(0x0018,0x0021),
00100  gdc.Tag(0x0018,0x0023),
00101  gdc.Tag(0x0018,0x0025),
00102  gdc.Tag(0x0018,0x0080),
00103  gdc.Tag(0x0018,0x0081),
00104  gdc.Tag(0x0018,0x0083),
00105  gdc.Tag(0x0018,0x0084),
00106  gdc.Tag(0x0018,0x0085),
00107  gdc.Tag(0x0018,0x0086),
00108  gdc.Tag(0x0018,0x0087),
00109  gdc.Tag(0x0018,0x0091),
00110  gdc.Tag(0x0018,0x0093),
00111  gdc.Tag(0x0018,0x0094),
00112  gdc.Tag(0x0018,0x0095),
00113  gdc.Tag(0x0018,0x1088),
00114  gdc.Tag(0x0018,0x1090),
00115  gdc.Tag(0x0018,0x1094),
00116  gdc.Tag(0x0018,0x1250),
00117  gdc.Tag(0x0018,0x1251),
00118  gdc.Tag(0x0018,0x1310),
00119  gdc.Tag(0x0018,0x1312),
00120  gdc.Tag(0x0018,0x1314),
00121  gdc.Tag(0x0018,0x1315),
00122  gdc.Tag(0x0018,0x1316),
00123  gdc.Tag(0x0020,0x0110),
00124  gdc.Tag(0x0028,0x0120),
00125  gdc.Tag(0x0028,0x1050),
00126  gdc.Tag(0x0028,0x1051)
00127  ]
00128  for tag in taglist:
00129      #print tag
00130      ano.Remove( tag )
00131
00132  # special handling
00133  gen = gdc.UIDGenerator()
00134  ano.Replace( gdc.Tag(0x0008,0x9123), gen.Generate() )
00135  #ano.Empty( gdc.Tag(0x0040,0x0555) )
00136
00137
00138  #
00139  # uid = gen.Generate()
00140  # de.SetTag( gdc.Tag(0x0008,0x0018) )
00141  # de.SetByteStringValue( uid )
00142  # ds.Insert( de )
00143
00144  # init FMI now:
00145  #fmi = f.GetHeader()

```

```

00146 #ts = gdcm.TransferSyntax()
00147 #print ts
00148 #fmi.SetDataSetTransferSyntax( ts ) # default
00149 #print fmi.GetDataSetTransferSyntax()
00150 #de.SetTag( gdcm.Tag(0x0002,0x0010) )
00151 #uid = "1.2.840.10008.1.2"
00152 #de.SetByteStringValue( uid )
00153 #fmi.Insert( de )
00154 # f.SetHeader( r.GetFile().GetHeader() )
00155
00156 writer = gdcm.Writer()
00157 writer.SetFile( ano.GetFile() )
00158 writer.SetFileName( "rawstorage.dcm" );
00159 writer.Write()

```

14.119 DecompressImage.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 if __name__ == "__main__":
00025
00026     file1 = sys.argv[1]
00027     file2 = sys.argv[2]
00028
00029     r = gdcm.ImageReader()
00030     r.SetFileName( file1 )
00031     if not r.Read():
00032         sys.exit(1)
00033
00034     # check GetFragment API:
00035     pd = r.GetFile().GetDataSet().GetDataElement(gdcm.Tag(0x7fe0, 0x0010))
00036     frags = pd.GetSequenceOfFragments();
00037     frags.GetFragment(0);
00038
00039     ir = r.GetImage()
00040     w = gdcm.ImageWriter()
00041     image = w.GetImage()
00042
00043     image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
00044     dims = ir.GetDimensions();
00045     print ir.GetDimension(0);
00046     print ir.GetDimension(1);
00047     print "Dims:",dims
00048
00049     # Just for fun:
00050     dircos = ir.GetDirectionCosines()
00051     t = gdcm.Orientation.GetType(tuple(dircos))
00052     l = gdcm.Orientation.GetLabel(t)
00053     print "Orientation label:",l
00054
00055     image.SetDimension(0, ir.GetDimension(0) );
00056     image.SetDimension(1, ir.GetDimension(1) );
00057
00058     pixeltype = ir.GetPixelFormat();
00059     image.SetPixelFormat( pixeltype );
00060
00061     pi = ir.GetPhotometricInterpretation();
00062     image.SetPhotometricInterpretation( pi );
00063
00064     pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
00065     str1 = ir.GetBuffer()
00066     #print ir.GetBufferLength()
00067     pixeldata.SetByteStringValue( str1 )
00068     image.SetDataElement( pixeldata )
00069
00070     w.SetFileName( file2 )
00071     w.SetFile( r.GetFile() )
00072     w.SetImage( image )

```

```
00073 if not w.Write():
00074     sys.exit(1)
```

14.120 DumbAnonymizer.py

```
00001
00014
00015 """
00016 This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
00017 This class becomes really handy when one knows which particular tag to fill in.
00018
00019 Usage:
00020
00021 python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm
00022
00023 """
00024
00025 import gdcm
00026
00027 # http://www.oid-info.com/get/1.3.6.1.4.17434
00028 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
00029
00030 tag_rules={
00031     # Value
00032     (0x0012,0x0010):("Value","MySponsorName"),
00033     (0x0012,0x0020):("Value","MyProtocolID"),
00034     (0x0012,0x0021):("Value","MyProtocolName"),
00035     (0x0012,0x0062):("Value","YES"),
00036     (0x0012,0x0063):("Value","MyDeidentificationMethod"),
00037
00038     # Method
00039     #(0x0002,0x0003):("Method","GenerateMSOPId"),
00040     #(0x0008,0x1155):("Method","GenerateMSOPId"),
00041     (0x0008,0x0018):("Method","GenerateMSOPId"),
00042     (0x0010,0x0010):("Method","GetSponsorInitials"),
00043     (0x0010,0x0020):("Method","GetSponsorId"),
00044     (0x0012,0x0030):("Method","GetSiteId"),
00045     (0x0012,0x0031):("Method","GetSiteName"),
00046     (0x0012,0x0040):("Method","GetSponsorId"),
00047     (0x0012,0x0050):("Method","GetTPId"),
00048     (0x0018,0x0022):("Method","KeepIfExist"),
00049     (0x0018,0x1315):("Method","KeepIfExist"),
00050     (0x0020,0x000d):("Method","GenerateStudyId"),
00051     (0x0020,0x000e):("Method","GenerateSeriesId"),
00052     (0x0020,0x1002):("Method","GetNumberOfFrames"),
00053     (0x0020,0x0020):("Method","GetPatientOrientation"),
00054     # Other:
00055     (0x0012,0x0051):("Patient Field","Type Examen"),
00056     (0x0018,0x1250):("Sequence Field","Receive Coil"),
00057     (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
00058     (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
00059     (0x0018,0x0082):("Sequence Field","Inversion Time"),
00060 }
00061
00062 class MyAnon:
00063     def __init__(self):
00064         self.studyuid = None
00065         self.seriesuid = None
00066         generator = gdcm.UIDGenerator()
00067         if not self.studyuid:
00068             self.studyuid = generator.Generate()
00069         if not self.seriesuid:
00070             self.seriesuid = generator.Generate()
00071     def GetSponsorInitials(self):
00072         return "dummy^foobar"
00073     def GenerateStudyId(self):
00074         return self.studyuid
00075     def GenerateSeriesId(self):
00076         return self.seriesuid
00077     #def GenerateMSOPId(self):
00078     def GenerateMSOPId(self):
00079         generator = gdcm.UIDGenerator()
00080         return generator.Generate()
00081     def GetSiteId(self):
00082         return "MySiteId"
00083     def GetSiteName(self):
00084         return "MySiteName"
```

```

00085 def GetSponsorId(self):
00086     return "MySponsorId"
00087 def GetTPId(self):
00088     return "MyTP"
00089
00090 if __name__ == "__main__":
00091     import sys
00092     gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle( "DumbAnonymizer" )
00093     gdcmm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
00094
00095     r = gdcmm.Reader()
00096     filename = sys.argv[1]
00097     r.SetFileName( filename )
00098     if not r.Read(): sys.exit(1)
00099
00100     obj = MyAnon()
00101
00102     w = gdcmm.Writer()
00103     ano = gdcmm.Anonymizer()
00104     ano.SetFile( r.GetFile() )
00105     ano.RemoveGroupLength()
00106     for tag,rule in tag_rules.items():
00107         if rule[0] == 'Value':
00108             print tag,rule
00109             ano.Replace( gdcmm.Tag( tag[0], tag[1] ), rule[1] )
00110         elif rule[0] == 'Method':
00111             print tag,rule
00112             # result = locals()[rule[1]]()
00113             methodname = rule[1]
00114             if hasattr(obj, methodname):
00115                 _member = getattr(obj, methodname)
00116                 result = _member()
00117                 ano.Replace( gdcmm.Tag( tag[0], tag[1] ), result )
00118             else:
00119                 print "Problem with: ", methodname
00120
00121     outfilename = sys.argv[2]
00122     w.SetFileName( outfilename )
00123     w.SetFile( ano.GetFile() )
00124     if not w.Write(): sys.exit(1)

```

14.121 ExtractImageRegion.py

```

00001
00014
00015 """
00016
00017 This small code shows how to use the gdcmm.ImageRegionReader API
00018 In this example we are taking each frame by frame and dump them to
00019 /tmp/frame.raw.
00020
00021 Usage:
00022 $ ExtractImageRegion.py input.dcm
00023
00024 Example:
00025 $ ExtractImageRegion.py gdcmmData/012345.002.050.dcm
00026 $ md5sum /tmp/frame.raw
00027 d594a5e2fde12f32b6633ca859b4d4a6 /tmp/frame.raw
00028 $ gdcminfo --md5sum gdcmmData/012345.002.050.dcm
00029 [...]
00030 md5sum: d594a5e2fde12f32b6633ca859b4d4a6
00031 """
00032
00033 import gdcmm
00034
00035 if __name__ == "__main__":
00036     import sys
00037     filename = sys.argv[1]
00038
00039     file_size = gdcmm.System.FileSize(filename);
00040
00041     # instantiate the reader:
00042     reader = gdcmm.ImageRegionReader();
00043     reader.SetFileName( filename );
00044
00045     # pull DICOM info:
00046     if not reader.ReadInformation():

```

```

00047     sys.exit(1)
00048
00049     # store current offset:
00050     cur_pos = reader.GetStreamCurrentPosition();
00051
00052     remaining = file_size - cur_pos;
00053
00054     print("Remaining bytes to read (Pixel Data): %d" % remaining );
00055
00056     # Get file infos
00057     f = reader.GetFile();
00058
00059     # get some info about image
00060     dims = gdcmm.ImageHelper.GetDimensionsValue(f);
00061     print(dims)
00062     pf = gdcmm.ImageHelper.GetPixelFormatValue (f);
00063     pixelsize = pf.GetPixelSize();
00064     pi = gdcmm.ImageHelper.GetPhotometricInterpretationValue(f);
00065     print( pi );
00066
00067     # buffer to get the pixels
00068     buffer = bytearray( dims[0] * dims[1] * pixelsize )
00069
00070     # define a simple box region.
00071     box = gdcmm.BoxRegion();
00072     for z in range(0, dims[2]):
00073         # Define that I want the image 0, full size (dimx x dimy pixels)
00074         # and do that for each z:
00075         box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
00076         #print( box.toString() );
00077         reader.SetRegion( box );
00078
00079         # reader will try to load the uncompressed image region into buffer.
00080         # the call returns an error when buffer.Length is too small. For instance
00081         # one can call:
00082         # uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
00083         # to get the exact size of minimum buffer
00084         if reader.ReadIntoBuffer(buffer):
00085             open('/tmp/frame.raw', 'wb').write(buffer)
00086         else:
00087             #throw new Exception("can't read pixels error");
00088             sys.exit(1)

```

14.122 FindAllPatientName.py

```

00001
00014 """
00015 This example shows how one can use the gdcmm.CompositeNetworkFunctions class
00016 for executing a C-FIND query
00017 It will print the list of patient name found
00018
00019 Usage:
00020
00021 python FindAllPatientName.py
00022
00023 """
00024
00025 import gdcmm
00026
00027 # Patient Name
00028 tag = gdcmm.Tag(0x10,0x10)
00029 de = gdcmm.DataElement(tag)
00030
00031 # Search all patient name where string match 'F*'
00032 de.SetByteStringValue('F*')
00033
00034 ds = gdcmm.DataSet()
00035 ds.Insert(de)
00036
00037 cnf = gdcmm.CompositeNetworkFunctions()
00038 theQuery = cnf.ConstructQuery (gdcmm.ePatientRootType,gdcmm.ePatient,ds)
00039
00040 #print theQuery.ValidateQuery()
00041
00042 # prepare the variable for output
00043 ret = gdcmm.DataSetArrayType()
00044

```

```

00045 # Execute the C-FIND query
00046 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
00047
00048 for i in range(0,ret.size()):
00049     print "Patient #",i
00050     print ret[i]

```

14.123 FixCommaBug.py

```

00001
00014
00015 """
00016 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
00017 "," as required by the DICOM standard
00018 Issue is still current (IMHO) with gdcms 2.0.9
00019 """
00020
00021 import gdcms
00022 import sys
00023
00024 filename = sys.argv[1]
00025 outname = sys.argv[2]
00026
00027 # read
00028 r = gdcms.Reader()
00029 r.SetFileName( filename )
00030 if not r.Read():
00031     print "not valid"
00032     sys.exit(1)
00033
00034 file = r.GetFile()
00035 dataset = file.GetDataSet()
00036
00037 ano = gdcms.Anonymizer()
00038 ano.SetFile( file )
00039
00040 tags = [
00041     gdcms.Tag(0x0018,0x1164),
00042     gdcms.Tag(0x0018,0x0088),
00043     gdcms.Tag(0x0018,0x0050),
00044     gdcms.Tag(0x0028,0x0030),
00045 ]
00046
00047 for tag in tags:
00048     print tag
00049     if dataset.FindDataElement( tag ):
00050         pixelpacing = dataset.GetDataElement( tag )
00051         #print pixelpacing
00052         bv = pixelpacing.GetByteValue()
00053         str = bv.GetBuffer()
00054         #print bv.GetLength()
00055         #print len(str)
00056         new_str = str.replace(",",".")
00057         # Need to explicitly pass bv.GetLength() to remove any trailing garbage
00058         ano.Replace( tag, new_str, bv.GetLength() )
00059
00060 #print dataset
00061
00062 w = gdcms.Writer()
00063 w.SetFile( file )
00064 w.SetFileName( outname )
00065 if not w.Write():
00066     print "Cannot write"
00067     sys.exit(1)
00068
00069 # paranoid:
00070 image_reader = gdcms.ImageReader()
00071 image_reader.SetFileName( outname )
00072 if not image_reader.Read():
00073     print "there is still a comma"
00074     sys.exit(1)
00075
00076 print "Success!"
00077 sys.exit(0) # success

```

14.124 GetPortionCSAHeader.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python GetPortionCSAHeader.py input.dcm
00019
00020 Footnote:
00021 SIEMENS is not publishing any information on the CSA header. So any info extracted
00022 is at your own risk.
00023 """
00024
00025 import sys
00026 import gdcm
00027
00028 if __name__ == "__main__":
00029
00030     file = sys.argv[1]
00031
00032     r = gdcm.Reader()
00033     r.SetFileName( file )
00034     if not r.Read():
00035         sys.exit(1)
00036
00037     ds = r.GetFile().GetDataSet()
00038     csa_t1 = gdcm.CSAHeader()
00039     csa_t2 = gdcm.CSAHeader()
00040     #print csa
00041     t1 = csa_t1.GetCSAImageHeaderInfoTag();
00042     print t1
00043     t2 = csa_t2.GetCSASeriesHeaderInfoTag();
00044     print t2
00045     # Let's do it for t1:
00046     if ds.FindDataElement( t1 ):
00047         csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
00048         print csa_t1
00049
00050     # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
00051     bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
00052     print bvalues
00053
00054     diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive !
00055     print diffgraddir
00056
00057     # repeat for t2 if you like it:
00058     if ds.FindDataElement( t2 ):
00059         csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
00060         # print csa_t2
00061
00062     gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
00063     print gdt
00064
00065     bv = gdt.GetByteValue();
00066     #print bv
00067     str = bv.GetPointer()
00068     print str.split("\\" )

```

14.125 HelloWorld.py

```

00001
00014
00015 """
00016 Hello World !
00017 """
00018
00019 import gdcm
00020 import sys
00021
00022 if __name__ == "__main__":
00023
00024     # verbosity:
00025     #gdcm.Trace.DebugOn()
00026     #gdcm.Trace.WarningOn()
00027     #gdcm.Trace.ErrorOn()

```



```

00028
00029 # Get the filename from the command line
00030 filename = sys.argv[1]
00031
00032 # Instantiate a gdcm.Reader
00033 # This is the main class to handle any type of DICOM object
00034 # You should check for gdcm.ImageReader for reading specifically DICOM Image file
00035 r = gdcm.Reader()
00036 r.SetFileName( filename )
00037 # If the reader fails to read the file, we should stop !
00038 if not r.Read():
00039     print "Not a valid DICOM file"
00040     sys.exit(1)
00041
00042 # Get the DICOM File structure
00043 file = r.GetFile()
00044
00045 # Get the DataSet part of the file
00046 dataset = file.GetDataSet()
00047
00048 # Ok let's print it !
00049 print dataset
00050
00051 # Use StringFilter to print a particular Tag:
00052 sf = gdcm.StringFilter()
00053 sf.SetFile(r.GetFile())
00054
00055 # Check if Attribute exist
00056 print dataset.FindElement( gdcm.Tag(0x0028,0x0010))
00057
00058 # Let's print it as string pair:
00059 print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))

```

14.126 ManipulateFile.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python ManipulateFile.py input.dcm output.dcm
00019
00020 Footnote:
00021 GDCM 1.2.x would create incorrect Multiframe MR Image Storage file. Try to recover from
00022 the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
00023 e.g:
00024
00025 python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
00026 """
00027
00028 import sys
00029 import gdcm
00030
00031 if __name__ == "__main__":
00032
00033     file1 = sys.argv[1]
00034     file2 = sys.argv[2]
00035
00036     r = gdcm.Reader()
00037     r.SetFileName( file1 )
00038     if not r.Read():
00039         sys.exit(1)
00040
00041     ano = gdcm.Anonymizer()
00042     ano.SetFile( r.GetFile() )
00043     ano.RemovePrivateTags()
00044     ano.Remove( gdcm.Tag(0x0032,0x1030) )
00045     ano.Remove( gdcm.Tag(0x008,0x14) )
00046     ano.Remove( gdcm.Tag(0x008,0x1111) )
00047     ano.Remove( gdcm.Tag(0x008,0x1120) )
00048     ano.Remove( gdcm.Tag(0x008,0x1140) )
00049     ano.Remove( gdcm.Tag(0x10,0x21b0) )
00050     ano.Empty( gdcm.Tag(0x10,0x10) )
00051     ano.Empty( gdcm.Tag(0x10,0x20) )
00052     ano.Empty( gdcm.Tag(0x10,0x30) )
00053     ano.Empty( gdcm.Tag(0x20,0x10) )
00054     ano.Empty( gdcm.Tag(0x32,0x1032) )

```

```

00055 ano.Empty( gdcm.Tag(0x32,0x1033) )
00056 ano.Empty( gdcm.Tag(0x40,0x241) )
00057 ano.Empty( gdcm.Tag(0x40,0x254) )
00058 ano.Empty( gdcm.Tag(0x40,0x253) )
00059 ano.Empty( gdcm.Tag(0x40,0x1001) )
00060 ano.Empty( gdcm.Tag(0x8,0x80) )
00061 ano.Empty( gdcm.Tag(0x8,0x50) )
00062 ano.Empty( gdcm.Tag(0x8,0x1030) )
00063 ano.Empty( gdcm.Tag(0x8,0x103e) )
00064 ano.Empty( gdcm.Tag(0x18,0x1030) )
00065 ano.Empty( gdcm.Tag(0x38,0x300) )
00066 g = gdcm.UIDGenerator()
00067 ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
00068 ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
00069 ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
00070 ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
00071 #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
00072 """
00073 ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
00074 ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
00075 ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
00076 ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
00077 ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
00078 ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
00079 ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
00080 ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
00081 ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
00082 ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
00083
00084 ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
00085 ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
00086 ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
00087 ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
00088
00089 ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
00090
00091 ano.Empty( gdcm.Tag(0x0020,0x0020) )
00092
00093 ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
00094
00095 #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
00096
00097 #ano.Empty( gdcm.Tag(0x0028,0x1052) ) #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
Intercept"/>
00098 #ano.Empty( gdcm.Tag(0x0028,0x1053) ) #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
Slope"/>
00099 #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" ) #<entry group="0028" element="1054" vr="LO" vm="1" name="Rescale
Type"/>
00100
00101 ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
00102 """
00103
00104 w = gdcm.Writer()
00105 w.SetFile( ano.GetFile() )
00106 w.SetFileName( file2 )
00107 if not w.Write():
00108     sys.exit(1)

```

14.127 ManipulateSequence.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python ManipulateSequence.py input.dcm output.dcm
00019
00020 This was tested using:
00021
00022 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
00023
00024 This is a dummy example on how to modify a value set in a nested-nested dataset
00025
00026 WARNING:
00027 Do not use as-is in production, this is just an example
00028 This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
00029 """

```

```

00030
00031 import sys
00032 import gdcm
00033
00034 if __name__ == "__main__":
00035
00036     file1 = sys.argv[1]
00037     file2 = sys.argv[2]
00038
00039     r = gdcm.Reader()
00040     r.SetFileName( file1 )
00041     if not r.Read():
00042         sys.exit(1)
00043
00044     f = r.GetFile()
00045     ds = f.GetDataSet()
00046     tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
00047     if ds.FindDataElement( tsis ):
00048         sis = ds.GetDataElement( tsis )
00049         #sqsis = sis.GetSequenceOfItems()
00050         # GetValueAsSQ handle more cases
00051         sqsis = sis.GetValueAsSQ()
00052         if sqsis.GetNumberOfItems():
00053             item1 = sqsis.GetItem(1)
00054             nestedds = item1.GetNestedDataSet()
00055             tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
00056             if nestedds.FindDataElement( tprcs ):
00057                 prcs = nestedds.GetDataElement( tprcs )
00058                 sqprcs = prcs.GetSequenceOfItems()
00059                 if sqprcs.GetNumberOfItems():
00060                     item2 = sqprcs.GetItem(1)
00061                     nestedds2 = item2.GetNestedDataSet()
00062                     # (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
00063                     tcm = gdcm.Tag(0x0008,0x0104)
00064                     if nestedds2.FindDataElement( tcm ):
00065                         cm = nestedds2.GetDataElement( tcm )
00066                         mystr = "GDCM was here"
00067                         cm.SetByteStringValue( mystr )
00068
00069     w = gdcm.Writer()
00070     w.SetFile( f )
00071     w.SetFileName( file2 )
00072     if not w.Write():
00073         sys.exit(1)

```

14.128 MergeFile.py

```

00001
00014 """
00015 """
00016 Usage:
00017
00018 python MergeFile.py input1.dcm input2.dcm
00019
00020 It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
00021 and copy the Stored Pixel values from input2.dcm
00022 This script even works when input2.dcm is a Secondary Capture and does not contains information
00023 such as IOP and IPP...
00024 """
00025
00026 import sys
00027 import gdcm
00028
00029 if __name__ == "__main__":
00030
00031     file1 = sys.argv[1]
00032     file2 = sys.argv[2]
00033
00034     r1 = gdcm.ImageReader()
00035     r1.SetFileName( file1 )
00036     if not r1.Read():
00037         sys.exit(1)
00038
00039     r2 = gdcm.ImageReader()
00040     r2.SetFileName( file2 )
00041     if not r2.Read():
00042         sys.exit(1)

```

```

00043
00044 # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
00045 # Instead always prefer to only copy the Raw Data Element.
00046 # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
00047 r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
00048
00049 w = gdcm.ImageWriter()
00050 w.SetFile( r1.GetFile() )
00051 #w.SetImage( r2.GetImage() ) # See comment above
00052 w.SetImage( r1.GetImage() )
00053
00054 w.SetFileName( "merge.dcm" )
00055 if not w.Write():
00056     sys.exit(1)
00057
00058 sys.exit(0)

```

14.129 NewSequence.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python NewSequence.py input.dcm output.dcm
00019
00020
00021 Thanks to Robert Irie for code
00022 """
00023
00024 import sys
00025 import gdcm
00026
00027 if __name__ == "__main__":
00028
00029     file1 = sys.argv[1]
00030     file2 = sys.argv[2]
00031
00032     r = gdcm.Reader()
00033     r.SetFileName( file1 )
00034     if not r.Read():
00035         sys.exit(1)
00036
00037     f = r.GetFile()
00038     ds = f.GetDataSet()
00039     #tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
00040
00041     # Create a dataelement
00042     de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
00043     de.SetByteStringValue("Occupation")
00044     de.SetVR(gdcm.VR(gdcm.VR.SH))
00045
00046     # Create an item
00047     it=gdcm.Item()
00048     it.SetVLToUndefined() # Needed to not popup error message
00049     #it.InsertDataElement(de)
00050     nds=it.GetNestedDataSet()
00051     nds.Insert(de)
00052
00053     # Create a Sequence
00054     sq=gdcm.SequenceOfItems().New()
00055     sq.SetLengthToUndefined()
00056     sq.AddItem(it)
00057
00058     # Insert sequence into data set
00059     des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
00060     des.SetVR(gdcm.VR(gdcm.VR.SQ))
00061     des.SetValue(sq.__ref__())
00062     des.SetVLToUndefined()
00063
00064     ds.Insert(des)
00065
00066     w = gdcm.Writer()
00067     w.SetFile( f )
00068     w.SetFileName( file2 )
00069     if not w.Write():
00070         sys.exit(1)

```

14.130 PhilipsPrivateRescaleInterceptSlope.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 filename = sys.argv[1]
00025 tmpfile = "/tmp/philips_rescaled.dcm"
00026
00027
00028 # Need to access some private tags, read the file :
00029 reader = gdcm.Reader()
00030 reader.SetFileName( filename )
00031 if not reader.Read():
00032     sys.exit(1)
00033
00034 ds = reader.GetFile().GetDataSet()
00035
00036 #print ds
00037 # (2005,1409)    DS      4      0.0
00038 # (2005,140a)    DS     16     1.52283272283272
00039
00040 # (2005,0014)    LO     26     Philips MR Imaging DD 005
00041 tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
00042 tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
00043 print tag1
00044 print tag2
00045
00046 # make sure to do a copy, we want the private tag to remain
00047 # otherwise gdcm gives us a reference
00048 el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
00049 print el1
00050 el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
00051 print el2
00052
00053 # (0028,1052) DS [-1000]                # 6, 1 RescaleIntercept
00054 # (0028,1053) DS [1]                   # 2, 1 RescaleSlope
00055
00056 el1.SetTag( gdcm.Tag(0x0028,0x1052) )
00057 el2.SetTag( gdcm.Tag(0x0028,0x1053) )
00058
00059 ds.Insert( el1 )
00060 ds.Insert( el2 )
00061
00062 w = gdcm.Writer()
00063 w.SetCheckFileMetaInformation( False )
00064 w.SetFileName( tmpfile )
00065 w.SetFile( reader.GetFile() )
00066 if not w.Write():
00067     sys.exit(1)
00068
00069 print "success"

```

14.131 PlaySound.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python PlaySound.py input.dcm
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
00025 filename = sys.argv[1]
00026 print filename

```

```

00027
00028 r = gdcm.Reader()
00029 r.SetFileName( filename )
00030 if not r.Read():
00031     sys.exit(1)
00032
00033 ds = r.GetFile().GetDataSet()
00034
00035 waveformtag = gdcm.Tag(0x5400,0x0100)
00036 waveformsq = ds.GetDataElement( waveformtag )
00037 #print waveformsq
00038
00039 #print dir(waveformsq)
00040
00041 items = waveformsq.GetSequenceOfItems()
00042
00043 if not items.GetNumberOfItems():
00044     sys.exit(1)
00045
00046 item = items.GetItem(1)
00047 #print item
00048
00049 waveformds = item.GetNestedDataSet()
00050 #print waveformds
00051
00052 waveformdatatag = gdcm.Tag(0x5400,0x1010)
00053 waveformdata = waveformds.GetDataElement( waveformdatatag )
00054
00055 #print waveformdata.GetPointer()
00056 bv = waveformdata.GetByteValue()
00057 print dir(bv)
00058
00059 #print bv.GetPointer()
00060 print bv.GetLength()
00061 l = 116838
00062
00063 file='test.wav'
00064 myfile = open(file, "wb")
00065 s = bv.GetPointer()
00066 for i in range(0, l):
00067     myfile.write(s[i])
00068 myfile.close()
00069
00070 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
00071 if sys.platform.startswith('win'):
00072     from winsound import PlaySound, SND_FILENAME, SND_ASYNC
00073     PlaySound(file, SND_FILENAME|SND_ASYNC)
00074 elif sys.platform.find('linux')>-1:
00075     from wave import open as waveOpen
00076     from ossaudiodev import open as ossOpen
00077     s = waveOpen(file,'rb')
00078     (nc,sw,fr,nf,comptype, compname) = s.getparams( )
00079     dsp = ossOpen('/dev/dsp','w')
00080     try:
00081         from ossaudiodev import AFMT_S16_NE
00082     except ImportError:
00083         if byteorder == "little":
00084             AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
00085         else:
00086             AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
00087     dsp.setparameters(AFMT_S16_NE, nc, fr)
00088     data = s.readframes(nf)
00089     s.close()
00090     dsp.write(data)
00091     dsp.close()

```

14.132 PrivateDict.py

```

00001
00014
00015 """
00016 """
00017
00018 import gdcm
00019 import sys,os
00020
00021 if __name__ == "__main__":

```

```

00022 #gdcmm.Trace.DebugOn()
00023 globInst = gdcmm.Global.GetInstance()
00024 # Try to load Part3.xml file
00025 # This file is too big for being accessible directly at runtime.
00026 globInst.LoadResourcesFiles()
00027
00028
00029 # Get a private tag from the runtime dicts. LoadResourcesFiles could
00030 # have failed but this has no impact on the private dict
00031
00032 d = globInst.GetDicts()
00033 print d.GetDictEntry( gdcmm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
00034 pd = d.GetPrivateDict()
00035 print pd.GetDictEntry( gdcmm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )

```

14.133 ReWriteSCAsMR.py

```

00001
00014
00015 """
00016 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale Slope/Intercept
00017 and saving the Pixel Spacing in (0028,0030)
00018 """
00019
00020 import gdcmm
00021 import sys,os
00022
00023 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
00024     ds = r.GetFile().GetDataSet()
00025     # Check Source Image Sequence
00026     if ds.FindDataElement( gdcmm.Tag(0x0008,0x2112) ):
00027         sis = ds.GetDataElement( gdcmm.Tag(0x0008,0x2112) )
00028         sqsis = sis.GetSequenceOfItems()
00029         if sqsis.GetNumberOfItems():
00030             item1 = sqsis.GetItem(1)
00031             nestedds = item1.GetNestedDataSet()
00032             if nestedds.FindDataElement( gdcmm.Tag(0x0008,0x1150) ):
00033                 ReferencedSOPClassUID = nestedds.GetDataElement( gdcmm.Tag(0x0008,0x1150) )
00034                 raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
00035                 uids = gdcmm.UIDs()
00036                 # what is the actual object we are looking at ?
00037                 ms = gdcmm.MediaStorage()
00038                 ms.SetFromDataSet(ds)
00039                 msuid = ms.GetString()
00040                 uids.SetFromUID( msuid )
00041                 msuidname = uids.GetName() # real Media Storage Name
00042                 uids.SetFromUID( raw )
00043                 sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
00044                 # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is correct
00045                 if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
00046                     return True
00047             # in all other case simply return the currentspacing:
00048             return False
00049
00050 if __name__ == "__main__":
00051     r = gdcmm.ImageReader()
00052     filename = sys.argv[1]
00053     r.SetFileName( filename )
00054     if not r.Read():
00055         sys.exit(1)
00056     f = r.GetFile()
00057
00058     if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
00059         # Special handling of the spacing:
00060         # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture Image Storage'
00061         # while we would rather have 'MR Image Storage'
00062         gdcmm.ImageHelper.SetForcePixelSpacing( True )
00063         mrspacing = gdcmm.ImageHelper.GetSpacingValue( r.GetFile() )
00064         # TODO: I cannot do simply the following:
00065         #image.SetSpacing( mrspacing )
00066         image.SetSpacing(0, mrspacing[0] )
00067         image.SetSpacing(1, mrspacing[1] )
00068         image.SetSpacing(2, mrspacing[2] )
00069         gdcmm.ImageHelper.SetForceRescaleInterceptSlope( True )
00070         ris = gdcmm.ImageHelper.GetRescaleInterceptSlopeValue( r.GetFile() )
00071         image.SetIntercept( ris[0] )
00072         image.SetSlope( ris[1] )

```

```

00073
00074 outfilename = sys.argv[2]
00075 w = gdcm.ImageWriter()
00076 w.SetFileName( outfilename )
00077 w.SetFile( r.GetFile() )
00078 w.SetImage( image )
00079 if not w.Write():
00080     sys.exit(1)
00081
00082 sys.exit(0)

```

14.134 ReadAndDumpDICOMDIR.py

```

00001
00023
00024
00025
00026 import sys
00027 import gdcm
00028
00029 if __name__ == "__main__":
00030     # Check arguments
00031     if (len(sys.argv) < 2):
00032         # No filename passed
00033         print "No input filename found"
00034         quit()
00035
00036     filename = sys.argv[1]
00037
00038
00039     # Read file
00040     reader = gdcm.Reader()
00041     reader.SetFileName(filename)
00042     if (not reader.Read()):
00043         print "Unable to read %s" % (filename)
00044         quit()
00045
00046     file = reader.GetFile()
00047
00048     # Retrieve header information
00049     fileMetaInformation = file.GetHeader()
00050     print fileMetaInformation
00051
00052     # Retrieve data set
00053     dataSet = file.GetDataSet()
00054     #print dataSet
00055
00056     # Check media storage
00057     mediaStorage = gdcm.MediaStorage()
00058     mediaStorage.SetFromFile(file)
00059     if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
00060         # File is not a DICOMDIR
00061         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
00062         quit()
00063
00064     # Check Media Storage SOP Class
00065     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
00066         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
00067         # Check SOP UID
00068         if (sopClassUid != "1.2.840.10008.1.3.10"):
00069             # File is not a DICOMDIR
00070             print "This file is not a DICOMDIR"
00071     else:
00072         # Not present
00073         print "Media Storage SOP Class not present"
00074         quit()
00075
00076     # Iterate through the DICOMDIR data set
00077     iterator = dataSet.GetDES().begin()
00078     while (not iterator.equal(dataSet.GetDES().end())):
00079         dataElement = iterator.next()
00080
00081         # Check the element tag
00082         if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
00083             # The 'Directory Record Sequence' element
00084             sequence = dataElement.GetValueAsSQ()
00085

```



```

00086     # Loop through the sequence items
00087     itemNr = 1
00088     while (itemNr < sequence.GetNumberOfItems()):
00089         item = sequence.GetItem(itemNr)
00090
00091         # Check the element tag
00092         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00093             # The 'Directory Record Type' element
00094             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00095
00096             # PATIENT
00097             while (value.strip() == "PATIENT"):
00098                 print value.strip()
00099                 # Print patient name
00100                 if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
00101                     value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
00102                     print value
00103
00104                 # Print patient ID
00105                 if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
00106                     value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
00107                     print value
00108
00109                 # Next
00110                 itemNr = itemNr + 1
00111                 item = sequence.GetItem(itemNr)
00112                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00113                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00114
00115             # STUDY
00116             while (value.strip() == "STUDY"):
00117                 print value.strip()
00118
00119                 # Print study UID
00120                 if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
00121                     value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
00122                     print value
00123
00124                 # Print study date
00125                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
00126                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
00127                     print value
00128
00129                 # Print study description
00130                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
00131                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
00132                     print value
00133
00134                 # Next
00135                 itemNr = itemNr + 1
00136                 item = sequence.GetItem(itemNr)
00137                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00138                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00139
00140             # SERIES
00141             while (value.strip() == "SERIES"):
00142                 print value.strip()
00143
00144                 # Print series UID
00145                 if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
00146                     value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).GetValue())
00147                     print value
00148
00149                 # Print series modality
00150                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
00151                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).GetValue())
00152                     print "Modality"
00153                     print value
00154
00155                 # Print series description
00156                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
00157                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).GetValue())
00158                     print "Description"
00159                     print value
00160
00161                 # Next
00162                 itemNr = itemNr + 1
00163                 item = sequence.GetItem(itemNr)
00164                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00165                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00166

```

```

00167             # IMAGE
00168             while (value.strip() == "IMAGE"):
00169                 print value.strip()
00170
00171             # Print image UID
00172             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
00173                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511)).GetValue())
00174                 print value
00175
00176             # Next
00177             if (itemNr < sequence.GetNumberOfItems()):
00178                 itemNr = itemNr + 1
00179             else:
00180                 break
00181
00182             item = sequence.GetItem(itemNr)
00183             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
00184                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
00185
00186             # Next
00187             itemNr = itemNr + 1

```

14.135 RemovePrivateTags.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python RemovePrivateTags.py input.dcm output.dcm
00019 """
00020
00021 import sys
00022 import gdcm
00023
00024
00025 if __name__ == "__main__":
00026
00027     file1 = sys.argv[1]
00028     file2 = sys.argv[2]
00029
00030     # Instantiate the reader.
00031     r = gdcm.Reader()
00032     r.SetFileName( file1 )
00033     if not r.Read():
00034         sys.exit(1)
00035
00036     # Remove private tags
00037     ano = gdcm.Anonymizer()
00038     ano.SetFile( r.GetFile() )
00039     if not ano.RemovePrivateTags():
00040         sys.exit(1)
00041
00042     # Write DICOM file
00043     w = gdcm.Writer()
00044     w.SetFile( ano.GetFile() )
00045     #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
00046     w.SetFileName( file2 )
00047     if not w.Write():
00048         sys.exit(1)
00049
00050     # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect) DICOM file
00051     # (application level)

```

14.136 ScanDirectory.py

```

00001
00014
00015 import gdcm
00016 import sys,os
00017
00018 class ProgressWatcher(gdcm.SimpleSubjectWatcher):

```

```

00019 def ShowProgress(self, sender, event):
00020     pe = gdcm.ProgressEvent.Cast(event)
00021     print pe.GetProgress()
00022 def EndFilter(self):
00023     print "Yay ! I am done"
00024
00025 if __name__ == "__main__":
00026     directory = sys.argv[1]
00027
00028     # Define the set of tags we are interested in
00029     t1 = gdcm.Tag(0x8,0x8);
00030     t2 = gdcm.Tag(0x10,0x10);
00031
00032     # Iterate over directory
00033     d = gdcm.Directory();
00034     nfiles = d.Load( directory );
00035     if(nfiles == 0): sys.exit(1);
00036     # System.Console.WriteLine( "Files:\n" + d.toString() );
00037
00038     filenames = d.GetFilenames()
00039
00040     # Get rid of any Warning while parsing the DICOM files
00041     gdcm.Trace.WarningOff()
00042
00043     # instantiate Scanner:
00044     sp = gdcm.Scanner.New();
00045     s = sp.__ref__()
00046     w = ProgressWatcher(s, 'Watcher')
00047
00048     s.AddTag( t1 );
00049     s.AddTag( t2 );
00050     b = s.Scan( filenames );
00051     if(not b): sys.exit(1);
00052
00053     print "success" ;
00054     #print s
00055
00056     pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
00057     pttv.Start()
00058     # iterate until the end:
00059     while( not pttv.IsAtEnd() ):
00060         # get current value for tag and associated value:
00061         # if tag was not found, then it was simply not added to the internal std::map
00062         # Warning value can be None
00063         tag = pttv.GetCurrentTag()
00064         value = pttv.GetCurrentValue()
00065         print tag,"->",value
00066         # increment iterator
00067         pttv.Next()
00068
00069     sys.exit(0)

```

14.137 SortImage.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python SortImage.py dirname
00019 """
00020
00021 import gdcm
00022 import sys
00023
00024 def PrintProgress(object, event):
00025     assert event == "ProgressEvent"
00026     print "Progress:", object.GetProgress()
00027
00028 def MySort(ds1, ds2):
00029     # compare ds1
00030     return False
00031
00032 if __name__ == "__main__":
00033
00034     dirname = sys.argv[1]
00035     d = gdcm.Directory()

```

```

00036 d.Load( dirname )
00037
00038 print d
00039
00040 sorter = gdcm.Sorter()
00041 sorter.SetSortFunction( MySort )
00042 #sorter.AddObserver( "ProgressEvent", PrintProgress )
00043 sorter.Sort( d.GetFileNames() )
00044
00045 print "Sorter:"
00046 print sorter

```

14.138 WriteBuffer.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 http://chuckhahm.com/Ischem/Zurich/XX_0134
00019
00020 (2005,1132) SQ (Sequence with undefined length #=8) # u/1, 1 Unknown Tag & Data
00021 (ffe,e000) na (Item with undefined length #=9) # u/1, 1 Item
00022 (2005,0011) LO [Philips MR Imaging DD 002] # 26, 1 PrivateCreator
00023 (2005,1137) PN [PDF_CONTROL_GEN_PARS] # 20, 1 Unknown Tag & Data
00024 (2005,1138) PN (no value available) # 0, 0 Unknown Tag & Data
00025 (2005,1139) PN [IEEE_PDF] # 8, 1 Unknown Tag & Data
00026 (2005,1140) PN (no value available) # 0, 0 Unknown Tag & Data
00027 (2005,1141) PN (no value available) # 0, 0 Unknown Tag & Data
00028 (2005,1143) SL 3103 # 4, 1 Unknown Tag & Data
00029 (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown Tag & Data
00030 (2005,1147) CS [Y] # 2, 1 Unknown Tag & Data
00031 (ffe,e00d) na (ItemDelimitationItem) # 0, 0 ItemDelimitationItem
00032 (ffe,e000) na (Item with undefined length #=9) # u/1, 1 Item
00033 (2005,0011) LO [Philips MR Imaging DD 002] # 26, 1 PrivateCreator
00034 (2005,1137) PN [PDF_CONTROL_PREP_PARS] # 22, 1 Unknown Tag & Data
00035 (2005,1138) PN (no value available) # 0, 0 Unknown Tag & Data
00036 (2005,1139) PN [IEEE_PDF] # 8, 1 Unknown Tag & Data
00037 (2005,1140) PN (no value available) # 0, 0 Unknown Tag & Data
00038 (2005,1141) PN (no value available) # 0, 0 Unknown Tag & Data
00039 (2005,1143) SL 7934 # 4, 1 Unknown Tag & Data
00040 (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown Tag & Data
00041 (2005,1147) CS [Y] # 2, 1 Unknown Tag & Data
00042 (ffe,e00d) na (ItemDelimitationItem) # 0, 0 ItemDelimitationItem
00043 ...
00044 """
00045
00046 import sys
00047 import gdcm
00048
00049 if __name__ == "__main__":
00050
00051 file1 = sys.argv[1]
00052 file2 = sys.argv[2]
00053
00054 r = gdcm.Reader()
00055 r.SetFileName( file1 )
00056 if not r.Read():
00057 sys.exit(1)
00058
00059 fg = gdcm.FileNameGenerator()
00060 f = r.GetFile()
00061 ds = f.GetDataSet()
00062 tsis = gdcm.Tag(0x2005,0x1132) #
00063 if ds.FindDataElement( tsis ):
00064 sis = ds.GetDataElement( tsis )
00065 #sqsis = sis.GetSequenceOfItems()
00066 # GetValueAsSQ handle more cases
00067 sqsis = sis.GetValueAsSQ()
00068 if sqsis.GetNumberOfItems():
00069 nitems = sqsis.GetNumberOfItems();
00070 fg.SetNumberOfFileNames( nitems )
00071 fg.SetPrefix( file2 )
00072 if not fg.Generate():
00073 print "problem"
00074 sys.exit(1)
00075 for i in range(0,nitems):

```

```

00076     item1 = sqsis.GetItem(i+1) # Item start at 1
00077     nestedds = item1.GetNestedDataSet()
00078     tprcs = gdcmm.Tag(0x2005,0x1144) #
00079     if nestedds.FindDataElement( tprcs ):
00080         prcs = nestedds.GetDataElement( tprcs )
00081         bv = prcs.GetByteValue()
00082         print bv
00083         f = open( fg.GetFilename(i) , "w" )
00084         f.write( bv.WriteBuffer() )

```

14.139 HelloActiviz.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcmm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcmm can be connected to Kitware.VTK Activiz product.
 * Three (3) arguments are required:
 * 1. Input DICOM file (SWIG)
 * 2. Temporary PNG (intermediate) file (Activiz)
 * 3. Final DICOM file (SWIG)
 *
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz.exe ~/Creatis/gdcmmData/test.acr out.png toto.dcm
 *
 * Footnote:
 * this test originally used vtkBMPWriter / vtkBMPReader combination to store intermediate
 * image file, but BMP file are 24bits by default. Instead use PNG format which supports seems
 * to be closer to what was expected in this simple test.
 */
public class HelloActiviz
{
    // Does not work with ActiViz.NET-5.4.0.455-Linux-x86_64-Personal
    /*
    static void ConnectSWIGToActiviz(Kitware.VTK.vtkImageExport imgin, Kitware.VTK.vtkImageImport imgout)
    {
        imgout.SetUpdateInformationCallback(imgin.GetUpdateInformationCallback());
        imgout.SetPipelineModifiedCallback(imgin.GetPipelineModifiedCallback());
        imgout.SetWholeExtentCallback(imgin.GetWholeExtentCallback());
        imgout.SetSpacingCallback(imgin.GetSpacingCallback());
        imgout.SetOriginCallback(imgin.GetOriginCallback());
        imgout.SetScalarTypeCallback(imgin.GetScalarTypeCallback());
        imgout.SetNumberOfComponentsCallback(imgin.GetNumberOfComponentsCallback());
        imgout.SetPropagateUpdateExtentCallback(imgin.GetPropagateUpdateExtentCallback());
        imgout.SetUpdateDataCallback(imgin.GetUpdateDataCallback());
        imgout.SetDataExtentCallback(imgin.GetDataExtentCallback());
        imgout.SetBufferPointerCallback(imgin.GetBufferPointerCallback());
        imgout.SetCallbackUserData(imgin.GetCallbackUserData());
    }
    */

    static Kitware.VTK.vtkImageData ConnectSWIGToActiviz(vtkgdcmm.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
        return imgout;
    }

    static vtkgdcmm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdcmm.vtkImageData imgout = new vtkgdcmm.vtkImageData( rawCppThis );
    }
}

```

```

    return imgout;
}

public static int Main(string[] args)
{
    string filename = args[0];
    string outfilename = args[1];

    // Step 1. Test SWIG -> Activiz
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.SetFileName( filename );
    //reader.Update(); // DO NOT call Update to check pipeline execution

    Kitware.VTK.vtkImageData imgout = ConnectSWIGToActiviz(reader.GetOutput());

    System.Console.WriteLine( imgout.ToString() ); // not initialized as expected

    vtkPNGWriter writer = new vtkPNGWriter();
    writer.SetInput( imgout );
    writer.SetFileName( outfilename );
    writer.Write();

    // Step 2. Test Activiz -> SWIG
    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );
    //bmpreader.Update(); // DO NOT update to check pipeline execution

    System.Console.WriteLine( bmpreader.GetOutput().ToString() ); // not initialized as expected

    vtkgdcml.vtkImageData imgout2 = ConnectActivizToSWIG(bmpreader.GetOutput());

    System.Console.WriteLine( imgout2.ToString() ); // not initialized as expected

    Kitware.VTK.vtkMedicalImageProperties prop = new Kitware.VTK.vtkMedicalImageProperties();
    prop.SetModality( "MR" );

    string outfilename2 = args[2];
    vtkGDCMImageWriter writer2 = vtkGDCMImageWriter.New();
    writer2.SetMedicalImageProperties( prop.CastToActiviz() );
    writer2.SetFileName( outfilename2 );
    writer2.SetInput( imgout2 );
    writer2.Write();

    return 0;
}
}

```

14.140 HelloActiviz2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * Usage:
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz2.exe gdcmlData/test.acr bla.png bla2.dcm
 */

/*
 * From the outside view, no-one can detect that object pass to/from
 * vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
 */

```

```

* TODO: Test Command/Observer
*/
public class HelloActiviz2
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];
        string outfilename2 = args[2];

        vtkGDCMImageReader reader = new Kitware.VTK.GDCM.vtkGDCMImageReader();
        reader.SetFileName( filename );

        // When calling multiple times creation of C# object from the same C++ object it triggers a:
        //error: potential refcounting error: Duplicate rawCppThis - weak reference that is still alive. Attempting to add '0x00b2dc10' again.
        //    Allowing new wrapped object to take over table key...
        //    Original object should *not* have been destroyed while we still had it in our table without notifying us...
        //reader.GetOutput();
        //reader.GetOutput();

        System.Console.WriteLine( reader.ToString() ); // Test the ToString compat with Activiz

        vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
        writer.SetInput( reader.GetOutput() );
        writer.SetFileName( outfilename2 );
        writer.Write();

        System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the ToString compat with Activiz

        System.Console.WriteLine( writer.ToString() ); // Test the ToString compat with Activiz

        vtkPNGWriter pngwriter = new vtkPNGWriter();
        pngwriter.SetInput( reader.GetOutput() );
        pngwriter.SetFileName( outfilename );
        pngwriter.Write();

        // at that point the .Write() should have triggered an Update() on the reader:
        if( reader.GetImageFormat() == vtkgdc.VTK_LUMINANCE ) // MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        vtkPNGReader bmpreader = new vtkPNGReader();
        bmpreader.SetFileName( outfilename );

        vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
        prop.SetModality( "MR" );

        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();
        writer2.SetFileName( outfilename2 );
        writer2.SetDirectionCosines( dircos );
        writer2.SetMedicalImageProperties( prop );
        writer2.SetInput( bmpreader.GetOutput() );
        writer2.Write();

        return 0;
    }
}

```

14.141 HelloActiviz3.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

```

```

using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz3
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
        reader.Update();

        //System.Console.Write(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer2 viewer = vtkImageViewer2.New();
        viewer.SetInput(reader.GetOutput());
        viewer.SetupInteractor(iren);
        viewer.SetSize(600, 600);
        viewer.Render();

        iren.Initialize();
        iren.Start();

        return 0;
    }
}

```

14.142 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz4
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
        reader.Update();

        //System.Console.Write(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer viewer = vtkImageViewer.New();
    }
}

```



```

viewer.SetInput(reader.GetOutput());
viewer.SetupInteractor(iren);
viewer.SetSize(600, 600);
viewer.Render();

iren.Initialize();
iren.Start();

return 0;
}
}

```

14.143 HelloActiviz5.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

// The command line arguments are:
// -I      => run in interactive mode; unless this is used, the program will
//         not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz5.exe -I
 */
public class HelloActiviz5
{
    public static int Main(string[] args)
    {
        vtkTesting testHelper = vtkTesting.New();
        for ( int cc = 0; cc < args.Length; cc++ )
        {
            //testHelper.AddArguments(argc,const_cast<const char **>(argv));
            //System.Console.Write( "args: " + args[cc] + "\n" );
            testHelper.AddArgument( args[cc] );
        }
        if ( testHelper.IsFlagSpecified("-D") != 0 )
        {
            string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
            if( VTK_DATA_ROOT != null )
            {
                //System.Console.Write( "VTK_DATA_ROOT: " + VTK_DATA_ROOT + "\n" );
                testHelper.SetDataRoot(VTK_DATA_ROOT);
                testHelper.AddArgument("-D");
                testHelper.AddArgument(VTK_DATA_ROOT);
            }
        }

        string dataRoot = testHelper.GetDataRoot();
        string filename = dataRoot;
        filename += "/Data/mr.001";

        vtkDirectory dir = vtkDirectory.New();
        if( dir.FileIsDirectory( dataRoot ) == 0 )
        {
            filename = vtkGDCMTesting.GetGDCMDataRoot() + "/test.acr";
        }
        //System.Console.Write( "dataRoot: " + dataRoot + "\n" );
        System.Console.Write( "filename being used is: " + filename + "\n" );

        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

```

```

reader.SetFileNames(array);
reader.Update();

System.Console.Write(reader.GetOutput());

vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

vtkRenderer ren1 = vtkRenderer.New();
vtkRenderWindow renWin = vtkRenderWindow.New();
renWin.AddRenderer(ren1);

vtkImageActor actor = vtkImageActor.New();

vtkImageMapToWindowLevelColors coronalColors = vtkImageMapToWindowLevelColors.New();
coronalColors.SetInput(reader.GetOutput());

actor.SetInput(coronalColors.GetOutput());

ren1.AddActor(actor);
iren.SetRenderWindow(renWin);

iren.Initialize();

renWin.Render();

int retVal = testHelper.IsInteractiveModeSpecified();

if( retVal != 0 )
{
    iren.Start();
}

return 0;
}
}

```

14.144 HelloVTKWorld.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcml;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        reader.SetFileName( filename );
        reader.Update();

        vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
        System.Console.WriteLine( prop.GetPatientName() ); //

        if( reader.GetImageFormat() == vtkgdcml.vtkgdcml.VTK_LUMINANCE ) // MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        // Just for fun, invert the direction cosines, output should reflect that:
        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        string outfilename = args[1];
    }
}

```

```

    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.Write();

    return 0;
}

```

14.145 HelloVTKWorld2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using vtkgdcms;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld2
{
    public static int Main(string[] args)
    {
        string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();

        vtkVolume16Reader reader = vtkVolume16Reader.New();
        reader.SetDataDimensions(64, 64);
        reader.SetDataByteOrderToLittleEndian();
        reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsqu/quarter");
        reader.SetImageRange(1, 93);
        reader.SetDataSpacing(3.2, 3.2, 1.5);

        vtkImageCast cast = vtkImageCast.New();
        cast.SetInputConnection( reader.GetOutputPort() );
        cast.SetOutputScalarTypeToUnsignedChar();

        // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
        vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
        writer.SetFileName( "headsqu.dcm" );
        writer.SetInputConnection( reader.GetOutputPort() );
        // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
        // writer.SetInputConnection( cast.GetOutputPort() );
        writer.SetFileDimensionality( 3 );
        writer.Write();

        return 0;
    }
}

```

14.146 MetaImageMD5Activiz.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.

```

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
using gdcm;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/MetaImageMD5Activiz.exe gdcmData/012345.002.050.dcm
 */
public class MetaImageMD5Activiz
{
    public static int ProcessOneMHDMD5(string filename)
    {
        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        reader.FileLowerLeftOn();
        reader.DebugOff();
        int canread = reader.CanReadFile( filename );
        if( canread == 0 )
        {
            string refms = gdcm.Testing.GetMediaStorageFromFile(filename);
            if( gdcm.MediaStorage.IsImage( gdcm.MediaStorage.GetMSType(refms) ) )
            {
                System.Console.Write( "Problem with file: " + filename + "\n" );
                return 1;
            }
            // not an image
            return 0;
        }

        reader.SetFileName( filename );
        reader.Update();

        // System.Console.Write(reader.GetOutput());

        vtkMetaImageWriter writer = vtkMetaImageWriter.New();
        writer.SetCompression( false );
        writer.SetInput( reader.GetOutput() );
        string subdir = "MetaImageMD5Activiz";
        string tmpdir = gdcm.Testing.GetTempDirectory( subdir );
        if( !gdcm.PosixEmulation.FileIsDirectory( tmpdir ) )
        {
            gdcm.PosixEmulation.MakeDirectory( tmpdir );
        }
        string mhdfile = gdcm.Testing.GetTempFilename( filename, subdir );

        string rawfile = mhdfile;
        mhdfile += ".mhd";
        rawfile += ".raw";
        writer.SetFileName( mhdfile );
        writer.Write();

        string digestmhd = gdcm.Testing.ComputeFileMD5( mhdfile );
        string digestraw = gdcm.Testing.ComputeFileMD5( rawfile );

        string mhdref = vtkGDCMTesting.GetMHDMD5FromFile(filename);
        string rawref = vtkGDCMTesting.GetRAWMD5FromFile(filename);

        if( mhdref != digestmhd )
        {
            System.Console.Write( "Problem with mhd file: " + filename + "\n" );
            System.Console.Write( digestmhd );
            System.Console.Write( "\n" );
            System.Console.Write( mhdref );
            System.Console.Write( "\n" );
            return 1;
        }
        if( rawref != digestraw )
        {
            System.Console.Write( "Problem with raw file: " + filename + "\n" );
            System.Console.Write( digestraw );
            System.Console.Write( "\n" );
            System.Console.Write( rawref );
            System.Console.Write( "\n" );
            return 1;
        }
    }
}

```

```

    }

    return 0;
}
public static int Main(string[] args)
{
    if ( args.Length == 1 )
    {
        string filename = args[0];
        return ProcessOneMHDMD5( filename );
    }
    // Loop over all gdcmData
    gdcm.Trace.DebugOff();
    gdcm.Trace.WarningOff();
    gdcm.Trace.ErrorOff();

    uint n = gdcm.Testing.GetNumberOfFileNames();
    int ret = 0;
    for( uint i = 0; i < n; ++i )
    {
        string filename = gdcm.Testing.GetFileName( i );
        ret += ProcessOneMHDMD5( filename );
    }
    return ret;
}
}

```

14.147 RefCounting.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * this is not so much an example but simply a test to make sure ctor / dtor work as expected
 * and call the ::New and ->Delete() of VTK style.
 */
public class RefCounting
{
    public static int Main(string[] args)
    {
        vtkGDCMTesting testing1 = vtkGDCMTesting.New();
        vtkGDCMTesting testing2 = new vtkGDCMTesting(); // just in case people do not read STYLE documentation

        vtkGDCMImageReader reader1 = vtkGDCMImageReader.New();
        vtkGDCMImageReader reader2 = new vtkGDCMImageReader();

        vtkGDCMImageWriter writer1 = vtkGDCMImageWriter.New();
        vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();

        using (vtkGDCMTesting testing3 = new vtkGDCMTesting())
        {
            System.Console.Write( "GetReferenceCount: " + testing1.GetReferenceCount() + "\n");
            System.Console.Write( "GetReferenceCount: " + testing2.GetReferenceCount() + "\n");
            System.Console.Write( "GetReferenceCount: " + testing3.GetReferenceCount() + "\n");
        }

        using (vtkGDCMImageReader reader3 = new vtkGDCMImageReader())
        {
            System.Console.Write( "GetReferenceCount: " + reader3.GetReferenceCount() + "\n");
        }

        using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter.New())
        {
            System.Console.Write( "GetReferenceCount: " + writer3.GetReferenceCount() + "\n");
        }
    }
}

```

```

    }

    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
}
}

```

14.148 Compute3DSpacing.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader2.h"
#include "vtkImageChangeInformation.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"
#include "gdcmIPPSorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#else
#define vtkFloatingPointType float
#endif

/*
 * Simple example to check computation of spacing within vtkGDCMImageReader2
 * This is a direct implementation of:
 *
 * http://gdcm.sourceforge.net/wiki/index.php/Using_GDCM_API#Automatic_ordering_of_slices_for_vtkGDCMImageReader.SetFileNames
 *
 * For more advanced information on how 3D spacing is being computed see:
 *
 * - http://gdcm.sourceforge.net/html/classgdcm_1_1IPPSorter.html
 *
 * Usage:
 *
 * $ Compute3DSpacing SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm \
 *   SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm
 */

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;

    std::vector<std::string> filenames;
    for( int i = 1; i < argc; ++i )
    {
        filenames.push_back( argv[i] );
    }

    gdcm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
    {
        std::cerr << "Failed to sort files" << std::endl;
        return 1;
    }
    std::cout << "Sorting succeeded:" << std::endl;
    //s.Print( std::cout );

    std::cout << "Found z-spacing:" << std::endl;
    std::cout << s.GetZSpacing() << std::endl;
}

```

```

const double ippzspacing = s.GetZSpacing();

const std::vector<std::string> & sorted = s.GetFileNames();
vtkGDCMImageReader2 * reader = vtkGDCMImageReader2::New();
vtkStringArray *files = vtkStringArray::New();
std::vector< std::string >::const_iterator it = sorted.begin();
for( ; it != sorted.end(); ++it)
{
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
}
reader->SetFileNames( files );
reader->Update();

const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#ifdef VTK_MAJOR_VERSION >= 6
v16->SetInputConnection( reader->GetOutputPort() );
#else
v16->SetInput( reader->GetOutput() );
#endif
v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
v16->Update();

v16->GetOutput()->Print( std::cout );

return 0;
}

```

14.149 Convert16BitsTo8Bits.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmsTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmsData/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcms::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/012345.002.050.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
#ifdef VTK_MAJOR_VERSION >= 6
    cast->SetInputConnection( reader->GetOutputPort() );
#else
    cast->SetInput( reader->GetOutput() );
#endif
    cast->SetOutputScalarTypeToUnsignedChar();

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();

```

```

writer->SetFileName( "/tmp/cast.dcm" );
#ifdef (VTK_MAJOR_VERSION >= 6)
writer->SetInputConnection( cast->GetOutputPort() );
#else
writer->SetInput( cast->GetOutput() );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->Write();

reader->Delete();
cast->Delete();
writer->Delete();

return 0;
}

```

14.150 ConvertMultiFrameToSingleFrame.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcTesting.h"
#include "gdcFilenameGenerator.h"

int main(int argc, char *argv[])
{
    std::string filename;
    if( argc <= 1 )
    {
        const char *directory = gdcTesting::GetDataRoot();
        if(!directory) return 1;
        std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
        filename = file;
    }
    else
    {
        filename = argv[1];
    }
    std::cout << "file: " << filename << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( filename.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    int dims[3];
    reader->GetOutput()->GetDimensions( dims );

    std::ostream os;
    os << "singleframe";
    os << "%04d.dcm";
    gdcFilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = dims[2];
    fg.SetNumberOfFiles( nfiles );
    bool b = fg.Generate();
    if( !b )

```



```

{
    std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
    return 1;
}
if( !fg.GetNumberOfFilenames() )
{
    std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
    return 1;
}

// By default write them as Secondary Capture (for portability)
vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
vtkStringArray *filenames = vtkStringArray::New();
for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++i)
{
    filenames->InsertNextValue( fg.GetFilename(i) );
}
assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFilenames() );
writer->SetFileNames( filenames );
filenames->Delete();
writer->SetFileDimensionality( 2 );
#ifdef VTK_MAJOR_VERSION >= 6
writer->SetInputConnection( reader->GetOutputPort() );
#else
writer->SetInput( reader->GetOutput() );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->Write();

reader->Delete();
writer->Delete();

return 0;
}

```

14.151 ConvertRGBToLuminance.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"
#include "vtkVersion.h"

#include "gdcmlTesting.h"

// There is no such thing as MR Image Storage + Photometric Interpretation = RGB
// let's rewrite that into a proper single component image:
int main(int, char *[])
{
    const char *directory = gdcml::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/SIEMENS-MR-RGB-16Bits.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageLuminance *luminance = vtkImageLuminance::New();
#ifdef VTK_MAJOR_VERSION >= 6
    luminance->SetInputConnection( reader->GetOutputPort() );
#else
    luminance->SetInput( reader->GetOutput() );

```

```

#endif

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/bla.dcm" );
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( luminance->GetOutputPort() );
    #else
        writer->SetInput( luminance->GetOutput() );
    #endif
    //writer->SetImageFormat( reader->GetImageFormat() ); // Do NOT pass image format
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->Write();

    // TODO:
    //vtkImageAppendComponents.h

    reader->Delete();
    luminance->Delete();
    writer->Delete();

    return 0;
}

```

14.152 ConvertSingleBitTo8Bits.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkPointData.h"
#include "vtkBitArray.h"
#include "vtkUnsignedCharArray.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( filename );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkDataArray* array = reader->GetOutput()->GetPointData()->GetScalars();
    vtkBitArray *barray = vtkBitArray::SafeDownCast( array );
    if( !barray ) return false;
    vtkIdType nvalues = array->GetNumberOfTuples();
    vtkUnsignedCharArray *uarray = vtkUnsignedCharArray::New();
    uarray->SetNumberOfTuples( nvalues );
    for(vtkIdType i = 0; i < nvalues; ++i)
    {
        uarray->SetValue( i, (unsigned char)barray->GetValue(i) );
    }

    vtkImageData *copy = vtkImageData::New();

```

```

// http://www.vtk.org/Wiki/VTK/VTK_6_Migration/Changes_to_Scalars_Manipulation_Functions#AllocateScalars.28.29
copy->SetExtent( reader->GetOutput()->GetExtent() );
#if ( VTK_MAJOR_VERSION >= 6)
copy->AllocateScalars(VTK_UNSIGNED_CHAR, 3);
#else
copy->SetScalarType( VTK_UNSIGNED_CHAR );
copy->AllocateScalars();
#endif

//uarray->Print( std::cout );
//copy->GetPointData()->GetScalars()->Print( std::cout );
copy->GetPointData()->SetScalars( uarray );
uarray->Delete();

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileName( outfilename );
//writer->SetInput( cast->GetOutput() );
#if ( VTK_MAJOR_VERSION >= 6)
writer->SetInputData( copy );
#else
writer->SetInput( copy );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->SetFileDimensionality( reader->GetFileDimensionality( ) );
writer->Write();

reader->Delete();
copy->Delete();
writer->Delete();

return 0;
}

```

14.153 CreateFakePET.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFilenameGenerator.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{

```

```

gdcmm::Trace::DebugOn();

const vtkIdType xSize = 512;
const vtkIdType ySize = 512;
const vtkIdType zSize = 512;

// Create the filenames in advance to supply to the vtkGDCMImageWriter
std::ostringstream os;
os << "PT";
os << "%03d.dcm";
gdcmm::FilenameGenerator fg;
fg.SetPattern( os.str().c_str() );
unsigned int nfiles = zSize;
fg.SetNumberOfFilesNames( nfiles );
bool b = fg.Generate();
if( !b )
{
    std::cerr << "FilenameGenerator::Generate() failed" << std::endl;
    return 1;
}
if( !fg.GetNumberOfFilesNames() )
{
    std::cerr << "FilenameGenerator::Generate() failed somehow..." << std::endl;
    return 1;
}

vtkStringArray *filenames = vtkStringArray::New();
for(unsigned int i = 0; i < fg.GetNumberOfFilesNames(); ++i)
{
    filenames->InsertNextValue( fg.GetFilename(i) );
}

vtkImageData *image = vtkImageData::New();
image->SetDimensions(xSize,ySize,zSize);
image->SetOrigin(-350.684,350.0,890.76);
image->SetSpacing(5.4688,-5.4688,-3.27);
#if VTK_MAJOR_VERSION <= 5
    image->SetNumberOfScalarComponents(1);
    image->SetScalarTypeToDouble();
#else
    image->AllocateScalars(VTK_DOUBLE,1);
#endif

double pt[3];
for( int z = 0; z < zSize; ++z )
    for( int y = 0; y < ySize; ++y )
        for( int x = 0; x < xSize; ++x )
        {
            pt[0] = x;
            pt[1] = y;
            pt[2] = z;
            pt[0] -= xSize / 2;
            pt[1] -= ySize / 2;
            pt[2] -= zSize / 2;
            pt[0] /= xSize / 2;
            pt[1] /= ySize / 2;
            pt[2] /= zSize / 2;
            const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
            const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
            double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
            pixel[0] = inval;
        }

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileDimensionality( 2 );
writer->SetFileNames(filenames);
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputData( image );
#else
    writer->SetInput( image );
#endif
writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
writer->GetMedicalImageProperties()->SetModality( "PT" );
writer->SetScale( 0.0042 ); // why not
writer->Write();

image->Delete();
writer->Delete();

return 0;

```

```
}

```

14.154 CreateFakeRTDOSE.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{
    //gdcm::Trace::DebugOn();

    const vtkIdType xSize = 512;
    const vtkIdType ySize = 512;
    const vtkIdType zSize = 512;

    vtkImageData *image = vtkImageData::New();
    image->SetDimensions(xSize,ySize,zSize);
    image->SetOrigin(-350.684,350.0,890.76);
    image->SetSpacing(5.4688,-5.4688,-3.27);
    #if VTK_MAJOR_VERSION <= 5
        image->SetNumberOfScalarComponents(1);
        image->SetScalarTypeToDouble();
    #else
        image->AllocateScalars(VTK_DOUBLE,1);
    #endif

    double pt[3];
    for( int z = 0; z < zSize; ++z )
        for( int y = 0; y < ySize; ++y )
            for( int x = 0; x < xSize; ++x )
            {
                pt[0] = x;
                pt[1] = y;
                pt[2] = z;
                pt[0] -= xSize / 2;
                pt[1] -= ySize / 2;
                pt[2] -= zSize / 2;
                pt[0] /= xSize / 2;
                pt[1] /= ySize / 2;
                pt[2] /= zSize / 2;
                const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
                const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
                double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
                pixel[0] = inval;
            }

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();

```

```

writer->SetFileDimensionality( 3 );
writer->SetFileName( "rtdose.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputData( image );
#else
writer->SetInput( image );
#endif
writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Units", "GY");
writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Summation Type", "PLAN");
writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Type", "PHYSICAL");
writer->GetMedicalImageProperties()->AddUserDefinedValue( "Frame of Reference UID",
    "1.3.12.2.1107.5.6.1.68100.30270111041215391275000000001");
writer->GetMedicalImageProperties()->SetModality( "RTDOSE" );
//writer->GetMedicalImageProperties()->SetModality( "PT" ); // debug
writer->SetScale( 0.0042 ); // why not
writer->Write();

image->Delete();
writer->Delete();

// BEGIN HACK
// In GDCM version 2.4.3 and before, the following tag was missing which caused issue with some RTDose software:

// Open the DICOM file that was temporarily created. This will allows me to used
// GDCM to append specific tags that allows the RTDOSE to be associated with the
// relevant CT images.
gdcm::Reader reader2;
reader2.SetFileName("rtdose.dcm" );
reader2.Read();
gdcm::File &file = reader2.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

// Required by some software and not automagically added by GDCM in old version
gdcm::Attribute<0x0028,0x0009> framePointer;
framePointer.SetNumberOfValues(1);
framePointer.SetValue( gdcm::Tag(0x3004,0x000C) );
ds.Replace( framePointer.GetAsDataElement() );

gdcm::Writer writer2;
writer2.CheckFileMetaInformationOff();
writer2.SetFileName("rtdose2.dcm");
writer2.SetFile( file );
writer2.Write();
// END HACK

return 0;
}

```

14.155 GenerateRTSTRUCT.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataWriter.h"
#include "vtkGDCMPolyDataReader.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRTStructSetProperties.h"
#include "vtkStringArray.h"
#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"

```

```

#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

#include <algorithm> //for std::find

#include "gdcmDirectoryHelper.h"

using namespace gdcm;

//view each organ independently of the others, to make sure that
//organ names correspond to actual segmentations.
void ShowOrgan(vtkPolyData* inData)
{
    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputData( inData );
    #else
        cubeMapper->SetInput( inData );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty *property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    vtkRenderer *renderer = vtkRenderer::New();
    vtkRenderWindow *renWin = vtkRenderWindow::New();
    renWin->AddRenderer(renderer);

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    renderer->AddActor(cubeActor);
    renderer->ResetCamera();
    renderer->SetBackground(1,1,1);

    renWin->SetSize(300,300);

    renWin->Render();
    iren->Start();

    cubeMapper->Delete();
    cubeActor->Delete();
    renderer->Delete();
    renWin->Delete();
    iren->Delete();
}

/*
 * Full application which ... RTSTRUCT
 */
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " directory-with-rtstruct-and-ct-images\n";
        return 1;
    }
    std::string theDirName(argv[1]);
    Directory::FileType theRTSeries =
        DirectoryHelper::GetRTStructSeriesUIDs(theDirName);

    gdcm::Directory theDir;
    theDir.Load(argv[1]);

    if (theRTSeries.empty())
    {
        std::cerr << "No RTStructs found for the test, ending." << std::endl;
        return 1;
    }

    for (size_t q = 0; q < theRTSeries.size(); q++)
    {

```

```

Directory::FileNamesType theRTNames =
    DirectoryHelper::GetFileNamesFromSeriesUIDs(theDirName, theRTSeries[q]);

if (theRTNames.empty()){
    std::cerr << "Unable to load RT Series " << theRTSeries[q] << ", continuing. " << std::endl;
    continue;
}

vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
reader->SetFileName( theRTNames[0].c_str() );
reader->Update();

//std::cout << reader->GetMedicalImageProperties()->GetStudyDate() << std::endl;

vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
int numMasks = reader->GetNumberOfOutputPorts() + 1; //add a blank one in
writer->SetNumberOfInputPorts( numMasks );
std::string thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + theRTSeries[q] + ".dcm";
gdcmm::Directory::FileNamesType theFileNames = theDir.GetFileNames();
//keep renaming the output until we get something that doesn't overwrite what was there already
int count = 0;
while (std::find(theFileNames.begin(), theFileNames.end(), thePotentialName) != theFileNames.end())
{
    char buff[255];
    snprintf(buff, sizeof(buff), "%d", count);
    thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + buff + "." + theRTSeries[q] + ".dcm";
}
writer->SetFileName( thePotentialName.c_str());
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
//this line is cheating, we won't have the same stuff, and may not have a struct
//to start with.
//have to go back to the original data to reconstruct the RTStructureSetProperties
//writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
//writer->Write();

//loop through the outputs in order to write them out as if they had been created and appended
vtkStringArray* roiNames = vtkStringArray::New();
vtkStringArray* roiAlgorithms = vtkStringArray::New();
vtkStringArray* roiTypes = vtkStringArray::New();
roiNames->SetNumberOfValues(numMasks);
roiAlgorithms->SetNumberOfValues(numMasks);
roiTypes->SetNumberOfValues(numMasks);
vtkAppendPolyData* append = vtkAppendPolyData::New();

//ok, now we'll add a blank organ
//the blank organ is to test to ensure that blank organs work; there have been crash reports
//this code is added at the beginning to ensure that the blank organs are read
//and preserved as individual organs.
vtkPolyData* blank = vtkPolyData::New();
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputData(0, blank);
#else
    writer->SetInput(0, blank);
#endif
roiNames->InsertValue(0, "blank");
roiAlgorithms->InsertValue(0, "blank");
roiTypes->InsertValue(0, "ORGAN");

//note the offsets used to place the rtstruct at the beginning of the newly generated RT.
//the idea is to run the program twice; first to generate an rtstruct with a blank mask (making
//sure that that functionality works), and then a second time to make sure that everything is
//being read properly. Multiple organs with the same name could cause some strangenesses.
for (int i = 1; i < numMasks; ++i)
{
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection(i, reader->GetOutputPort(i-1));
        append->AddInputConnection(reader->GetOutputPort(i-1));
    #else
        writer->SetInput(i, reader->GetOutput(i-1));
        append->AddInput(reader->GetOutput(i-1));
    #endif
    std::string theString = reader->GetRTStructSetProperties()->GetStructureSetROIName(i-1);
    roiNames->InsertValue(i, theString);
    theString = reader->GetRTStructSetProperties()->GetStructureSetROIGenerationAlgorithm(i-1);
    roiAlgorithms->InsertValue(i, theString);
    theString = reader->GetRTStructSetProperties()->GetStructureSetRTROIInterpretedType(i-1);
    roiTypes->InsertValue(i, theString);

    ShowOrgan(reader->GetOutput(i-1));
}

```



```

    }

    vtkRTStructSetProperties* theProperties = vtkRTStructSetProperties::New();
    writer->SetRTStructSetProperties(theProperties);
    writer->InitializeRTStructSet(theDirName,
        reader->GetRTStructSetProperties()->GetStructureSetLabel(),
        reader->GetRTStructSetProperties()->GetStructureSetName(),
        roiNames, roiAlgorithms, roiTypes);

    writer->SetRTStructSetProperties(theProperties);
    writer->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    reader->Delete();
    append->Delete();
    roiNames->Delete();
    roiTypes->Delete();
    theProperties->Delete();
    roiAlgorithms->Delete();
    blank->Delete();

    writer->Delete();
}
return 0;
}

```

14.156 MagnifyFile.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
#include "gdcmSystem.h"

// This is a simple test to magnify an image that is known to give excellent
// compression ratio. This will be our test for those large image
int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/test.acr";
    std::cout << file << std::endl;
    if( !gdcm::System::FileExists( file.c_str() ) ) return 1;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    #if (VTK_MAJOR_VERSION >= 6)
    cast->SetInputConnection( reader->GetOutputPort() );
    #else
    cast->SetInput( reader->GetOutput() );
    #endif
}

```

```

cast->SetOutputScalarTypeToUnsignedShort();

vtkImageMagnify *magnify = vtkImageMagnify::New();
#if (VTK_MAJOR_VERSION >= 6)
magnify->SetInputConnection( cast->GetOutputPort() );
#else
magnify->SetInput( cast->GetOutput() );
#endif
magnify->SetInterpolate( 1 );
magnify->SetInterpolate( 0 );
int factor = 100;
magnify->SetMagnificationFactors (factor, factor, 1);

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputConnection( magnify->GetOutputPort() );
#else
writer->SetInput( magnify->GetOutput() );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->Write();

// TODO:
//vtkImageAppendComponents.h

reader->Delete();
magnify->Delete();
writer->Delete();

return 0;
}

```

14.157 gdcmortoplanes.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"

```

```

#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmsystem.h"
#include "gdcmdirectory.h"
#include "gdcmlppsorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

//-----
class vtkOrthoPlanesCallback : public vtkCommand
{
public:
    static vtkOrthoPlanesCallback *New()
    { return new vtkOrthoPlanesCallback; }

    void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                  void *callData )
    {
        vtkImagePlaneWidget* self =
            reinterpret_cast< vtkImagePlaneWidget* >( caller );
        if(!self) return;

        double* wl = static_cast<double*>( callData );

        if ( self == this->WidgetX )
        {
            this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
        }
        else if( self == this->WidgetY )
        {
            this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
        }
        else if (self == this->WidgetZ)
        {
            this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
            this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
        }
    }

    vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}

    vtkImagePlaneWidget* WidgetX;
    vtkImagePlaneWidget* WidgetY;
    vtkImagePlaneWidget* WidgetZ;
};

int main( int argc, char *argv[] )
{
    //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv, "Data/headsq/quarter");

    //vtkVolume16Reader* v16 = vtkVolume16Reader::New();
    // v16->SetDataDimensions( 64, 64);
    // v16->SetDataByteOrderToLittleEndian();
    // v16->SetImageRange( 1, 93);
    // v16->SetDataSpacing( 3.2, 3.2, 1.5);
    // v16->SetFilePrefix( fname );
    // v16->SetDataMask( 0x7fff);
    // v16->Update();
    std::vector<std::string> filenames;
    if( argc < 2 )
    {
        std::cerr << argv[0] << " filename1.dcm [filename2.dcm ...]\n";
        return 1;
    }
    else
    {
        // Is it a single directory ? If so loop over all files contained in it:
        const char *filename = argv[1];

```

```

if( argc == 2 && gdcmm::System::FileIsDirectory( filename ) )
{
    std::cout << "Loading directory: " << filename << std::endl;
    bool recursive = false;
    gdcmm::Directory d;
    d.Load(filename, recursive);
    gdcmm::Directory::FileNamesType const &files = d.GetFilesNames();
    for( gdcmm::Directory::FileNamesType::const_iterator it = files.begin(); it != files.end(); ++it )
    {
        filenames.push_back( it->c_str() );
    }
}
else // list of files passed directly on the cmd line:
    // discard non-existing or directory
{
    for(int i=1; i < argc; ++i)
    {
        filename = argv[i];
        if( gdcmm::System::FileExists( filename ) )
        {
            if( gdcmm::System::FileIsDirectory( filename ) )
            {
                std::cerr << "Discarding directory: " << filename << std::endl;
            }
            else
            {
                filenames.push_back( filename );
            }
        }
        else
        {
            std::cerr << "Discarding non existing file: " << filename << std::endl;
        }
    }
}
//names->Print( std::cout );
}

```

```

vtkGDCMImageReader * reader = vtkGDCMImageReader::New();
double ippzspacing;
if( filenames.size() > 1 )
{
    //gdcmm::Trace::DebugOn();
    //gdcmm::Trace::WarningOn();
    gdcmm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
    {
        std::cerr << "Failed to sort files" << std::endl;
        return 1;
    }
    std::cout << "Sorting succeeded:" << std::endl;
    s.Print( std::cout );

    std::cout << "Found z-spacing:" << std::endl;
    std::cout << s.GetZSpacing() << std::endl;
    ippzspacing = s.GetZSpacing();

    const std::vector<std::string> & sorted = s.GetFilesNames();
    vtkStringArray *files = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it )
    {
        const std::string &f = *it;
        files->InsertNextValue( f.c_str() );
    }
    reader->SetFileNames( files );
    //reader->SetFileLowerLeft( 1 );
    reader->Update(); // important
    files->Delete();
}
else
{
    reader->SetFileName( argv[1] );
    reader->Update(); // important
    ippzspacing = reader->GetOutput()->GetSpacing()[2];
    ippzspacing = 4;
}

```

```

    }

    //reader->GetOutput()->Print( std::cout );
    //vtkFloatingPointType range[2];
    //reader->GetOutput()->GetScalarRange(range);
    //std::cout << "Range: " << range[0] << " " << range[1] << std::endl;

    const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();

    vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
    #if (VTK_MAJOR_VERSION >= 6)
    v16->SetInputConnection( reader->GetOutputPort() );
    #else
    v16->SetInput( reader->GetOutput() );
    #endif
    v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
    v16->Update();

    #if 0
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetInput( v16->GetOutput() );
    writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->SetFileName( "out.dcm" );
    writer->Write();
    #endif

    vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

    vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());

    vtkActor* outlineActor = vtkActor::New();
    outlineActor->SetMapper( outlineMapper);

    vtkRenderer* ren1 = vtkRenderer::New();
    vtkRenderer* ren2 = vtkRenderer::New();

    vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);

    vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

    vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);

    vtkProperty* ipwProp = vtkProperty::New();
    //assign default props to the ipw's texture plane actor

    vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
    #if (VTK_MAJOR_VERSION >= 6)
    planeWidgetX->SetInputConnection(v16->GetOutputPort());
    #else
    planeWidgetX->SetInput(v16->GetOutput());
    #endif
    #endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();
    planeWidgetX->On();
    planeWidgetX->InteractionOff();
    planeWidgetX->InteractionOn();

    vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetY->SetInteractor( iren);

```

```

planeWidgetY->SetKeyPressActivationValue('y');
planeWidgetY->SetPicker(picker);
planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
planeWidgetY->SetTexturePlaneProperty(ipwProp);
planeWidgetY->TextureInterpolateOn();
planeWidgetY->SetResliceInterpolateToLinear();
#ifdef (VTK_MAJOR_VERSION >= 6)
planeWidgetY->SetInputConnection(v16->GetOutputPort());
#else
planeWidgetY->SetInput(v16->GetOutput());
#endif
planeWidgetY->SetPlaneOrientationToYAxes();
//planeWidgetY->SetSlicePosition(102.4);
planeWidgetY->SetLookupTable( planeWidgetX->GetLookupTable());
planeWidgetY->DisplayTextOn();
planeWidgetY->UpdatePlacement();
planeWidgetY->On();

vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
planeWidgetZ->SetInteractor( iren);
planeWidgetZ->SetKeyPressActivationValue('z');
planeWidgetZ->SetPicker(picker);
planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
planeWidgetZ->SetTexturePlaneProperty(ipwProp);
planeWidgetZ->TextureInterpolateOn();
planeWidgetZ->SetResliceInterpolateToCubic();
#ifdef (VTK_MAJOR_VERSION >= 6)
planeWidgetZ->SetInputConnection(v16->GetOutputPort());
#else
planeWidgetZ->SetInput(v16->GetOutput());
#endif
planeWidgetZ->SetPlaneOrientationToZAxes();
//planeWidgetZ->SetSliceIndex(25);
planeWidgetZ->SetLookupTable( planeWidgetX->GetLookupTable());
planeWidgetZ->DisplayTextOn();
planeWidgetZ->On();

vtkImageOrthoPlanes *orthoPlanes = vtkImageOrthoPlanes::New();
orthoPlanes->SetPlane(0, planeWidgetX);
orthoPlanes->SetPlane(1, planeWidgetY);
orthoPlanes->SetPlane(2, planeWidgetZ);
orthoPlanes->ResetPlanes();

vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
cbk->WidgetX = planeWidgetX;
cbk->WidgetY = planeWidgetY;
cbk->WidgetZ = planeWidgetZ;
planeWidgetX->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
planeWidgetY->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
planeWidgetZ->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
cbk->Delete();

double wl[2];
planeWidgetZ->GetWindowLevel(wl);

// Add a 2D image to test the GetReslice method
//
vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
colorMap->PassAlphaToOutputOff();
colorMap->SetActiveComponent(0);
colorMap->SetOutputFormatToLuminance();
#ifdef (VTK_MAJOR_VERSION >= 6)
colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
#else
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
#endif
colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

vtkImageActor* imageActor = vtkImageActor::New();
imageActor->PickableOff();
#ifdef (VTK_MAJOR_VERSION >= 6)
imageActor->SetInputData(colorMap->GetOutput());
#else
imageActor->SetInput(colorMap->GetOutput());
#endif

// Add the actors
//
ren1->AddActor( outlineActor);

```

```

ren2->AddActor( imageActor);

ren1->SetBackground( 0.1, 0.1, 0.2);
ren2->SetBackground( 0.2, 0.1, 0.2);

renWin->SetSize( 600, 350);

ren1->SetViewport(0,0,0.58333,1);
ren2->SetViewport(0.58333,0,1,1);

// Set the actors' positions
//
renWin->Render();
//iren->SetEventPosition( 175,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetEventPosition( 475,175);
//iren->SetKeyCode('r');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//renWin->Render();

//ren1->GetActiveCamera()->Elevation(110);
//ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
//ren1->GetActiveCamera()->Azimuth(45);
//ren1->GetActiveCamera()->Dolly(1.15);
ren1->ResetCameraClippingRange();

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText( "R" );
cube->SetXMinusFaceText( "L" );
cube->SetYPlusFaceText( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText( "H" );
cube->SetZMinusFaceText( "F" );
cube->SetFaceTextScale( 0.666667 );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkMatrix4x4 *invert = vtkMatrix4x4::New();
invert->DeepCopy( reader->GetDirectionCosines() );
invert->Invert();

// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(90);
transform->Concatenate(invert);
axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform );

axes2->SetTotalLength( 1.5, 1.5, 1.5 );
axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
axes2->SetConeRadius ( 1.025 * axes2->GetConeRadius() );
axes2->SetSphereRadius ( 1.500 * axes2->GetSphereRadius() );

vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
    GetCaptionTextProperty();
tprop->ItalicOn();
tprop->ShadowOn();
tprop->SetFontFamilyToTimes();

axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

// Playback recorded events
//

```

```

//vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
//recorder->SetInteractor(iren);
//recorder->ReadFromInputStringOn();
//recorder->SetInputString(IOEventLog);

// Interact with data
// Render the image
//
iren->Initialize();
renWin->Render();

// Test SetKeyPressActivationValue for one of the widgets
//
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);

//int retVal = vtkRegressionTestImage( renWin );
//
//if ( retVal == vtkRegressionTester::DO_INTERACTOR)
//{
//    iren->Start();
//}

// Clean up
//
//recorder->Off();
//recorder->Delete();

ipwProp->Delete();
orthoPlanes->Delete();
planeWidgetX->Delete();
planeWidgetY->Delete();
planeWidgetZ->Delete();
colorMap->Delete();
imageActor->Delete();
picker->Delete();
outlineActor->Delete();
outlineMapper->Delete();
outline->Delete();
iren->Delete();
renWin->Delete();
ren1->Delete();
ren2->Delete();
v16->Delete();
reader->Delete();

return 0;
}

```

14.158 gdcmreslice.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"

```



```

#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( argv[1] );
    //reader->FileLowerLeftOn();
    reader->Update();

    vtkImageFlip *flip = vtkImageFlip::New();
    #if ( VTK_MAJOR_VERSION >= 6 )
        flip->SetInputConnection(reader->GetOutputPort());
    #else
        flip->SetInput(reader->GetOutput());
    #endif
    flip->SetFilteredAxis(0);
    flip->Update();

    vtkImageReslice *reslice = vtkImageReslice::New();
    //reslice->SetInput(reader->GetOutput());
    #if ( VTK_MAJOR_VERSION >= 6 )
        reslice->SetInputConnection(flip->GetOutputPort());
    #else
        reslice->SetInput(flip->GetOutput());
    #endif
    //reslice->SetResliceAxesDirectionCosines()
    reader->GetDirectionCosines()->Print(std::cout);
    vtkMatrix4x4 *invert = vtkMatrix4x4::New();
    invert->DeepCopy( reader->GetDirectionCosines() );
    invert->Invert();

    //reslice->SetResliceAxes( reader->GetDirectionCosines() );
    reslice->SetResliceAxes( invert );
    reslice->Update();
    vtkImageData* ima = reslice->GetOutput();

    vtkLookupTable* table = vtkLookupTable::New();
    table->SetNumberOfColors(1000);
    table->SetTableRange(0,1000);
    table->SetSaturationRange(0,0);
    table->SetHueRange(0,1);
    table->SetValueRange(0,1);
    table->SetAlphaRange(1,1);
    table->Build();

    // Texture
    vtkTexture* texture = vtkTexture::New();
    #if ( VTK_MAJOR_VERSION >= 6 )
        texture->SetInputData(ima);
    #else
        texture->SetInput(ima);
    #endif
    texture->InterpolateOn();
    texture->SetLookupTable(table);

    // PlaneSource
    vtkPlaneSource* plane = vtkPlaneSource::New();

    // PolyDataMapper
    vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
    #if ( VTK_MAJOR_VERSION >= 6 )
        planeMapper->SetInputConnection(plane->GetOutputPort());
    #else
        planeMapper->SetInput(plane->GetOutput());
    #endif

    // Actor
    vtkActor* planeActor = vtkActor::New();
    planeActor->SetTexture(texture);
    planeActor->SetMapper(planeMapper);

```

```

planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

// DICOM is RAH:
vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkTransform *transform = vtkTransform::New();
transform->Identity();
//reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(invert);
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

renwin->Render();
iren->Start();

// Clean up:
reader->Delete();
table->Delete();
texture->Delete();
plane->Delete();
planeMapper->Delete();
planeActor->Delete();
ren->Delete();
renwin->Delete();
iren->Delete();

return 0;
}

```

14.159 gdcmrntionplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"

```

```

#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmmReader.h"
#include "gdcmmAttribute.h"

/*
This example is just for fun. We found a RT Ion Plan Storage and simply extracted the viz stuff for VTK

RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    const char * outfilename2 = argv[3];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcmm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcmm::MediaStorage::RTIonPlanStorage )
    {
        return 1;
    }

    /*
(300a,03a2) SQ # u/1,1 Ion Beam Sequence
  (ffe,e000) na (Item with undefined length)
    (0008,1040) LO [Test] # 4,1 Institutional Department Name
    (300a,00b2) SH (no value) # 0,1 Treatment Machine Name
    (300a,00b3) CS [MU] # 2,1 Primary Dosimeter Unit
    (300a,00c0) IS [1] # 2,1 Beam Number
    (300a,00c2) LO [1] # 2,1 Beam Name
    (300a,00c4) CS [STATIC] # 6,1 Beam Type
    (300a,00c6) CS [PROTON] # 6,1 Radiation Type
    (300a,00ce) CS [TREATMENT] # 10,1 Treatment Delivery Type
    (300a,00d0) IS [0] # 2,1 Number of Wedges
    (300a,00e0) IS [1] # 2,1 Number of Compensators
    (300a,00ed) IS [0] # 2,1 Number of Boli
    (300a,00f0) IS [1] # 2,1 Number of Blocks
    (300a,0110) IS [2] # 2,1 Number of Control Points
    (300a,02ea) SQ # u/1,1 Ion Range Compensator Sequence
      (ffe,e000) na (Item with undefined length)
        (300a,00e1) SH [lucite] # 6,1 Material ID
        (300a,00e4) IS [1] # 2,1 Compensator Number
        (300a,00e5) SH [75hdhe5] # 8,1 Compensator ID
        (300a,00e7) IS [35] # 2,1 Compensator Rows
        (300a,00e8) IS [37] # 2,1 Compensator Columns
        (300a,00e9) DS [3.679991\4.249288] # 18,2 Compensator Pixel Spacing
        (300a,00ea) DS [-76.00\62.50] # 12,2 Compensator Position
        (300a,00ec) DS
          [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.88\44.88]
          # 7618,1-n Compensator Thickness Data
        (300a,02e0) CS [ABSENT] # 6,1 Compensator Divergence
        (300a,02e1) CS [SOURCE_SIDE] # 12,1 Compensator Mounting Position
        (300a,02e4) FL 39.2 # 4,1 Isocenter to Compensator Tray Distance
        (300a,02e5) FL 2.12 # 4,1 Compensator Column Offset
        (300a,02e8) FL 4.76 # 4,1 Compensator Milling Tool Diameter
      (ffe,e00d)
    */
    const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

```

```

gdcmm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x02ea);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
const gdcmm::Item & item2 = ssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcmm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
//      (300a,00e7) IS [35]                                # 2,1 Compensator Rows
gdcmm::Attribute<0x300a,0x00e7> at1;
const gdcmm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
//      (300a,00e8) IS [37]                                # 2,1 Compensator Columns
gdcmm::Attribute<0x300a,0x00e8> at2;
const gdcmm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
at2.SetFromDataElement( compensatorcols );
std::cout << at2.GetValue() << std::endl;

//      (300a,00e9) DS [3.679991\4.249288 ]                # 18,2 Compensator Pixel Spacing
gdcmm::Attribute<0x300a,0x00e9> at3;
const gdcmm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
at3.SetFromDataElement( compensatorpixelspacing );
std::cout << at3.GetValue(0) << std::endl;
//      (300a,00ea) DS [-76.00\62.50]                      # 12,2 Compensator Position
gdcmm::Attribute<0x300a,0x00ea> at4;
const gdcmm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
at4.SetFromDataElement( compensatorposition );
std::cout << at4.GetValue(0) << std::endl;

vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

vtkImageData *img = vtkImageData::New();
img->Initialize();
img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
//imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif

```

```

img->GetPointData()->SetScalars(d);

#if (VTK_MAJOR_VERSION >= 6)
#else
    img->Update();
#endif
img->Print(std::cout);

vtkXMLImageDataReader *writeb= vtkXMLImageDataReader::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
writeb->SetFileName( outfilename );
writeb->Write( );
/*
(300a,03a6) SQ                                # u/1,1 Ion Block Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [brass ]                        # 6,1 Material ID
(300a,00f7) FL 95.03                          # 4,1 Isocenter to Block Tray Distance
(300a,00f8) CS [APERTURE]                     # 8,1 Block Type
(300a,00fa) CS [ABSENT]                       # 6,1 Block Divergence
(300a,00fb) CS [SOURCE_SIDE ]                # 12,1 Block Mounting Position
(300a,00fc) IS [1 ]                          # 2,1 Block Number
(300a,0100) DS [50.00 ]                      # 6,1 Block Thickness
(300a,0104) IS [179 ]                        # 4,1 Block Number of Points
(300a,0106) DS
[1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44.2\29.7\43
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
(fffe,e00d)
(fffe,e0dd)

*/
gdcmm::Tag tblocksq(0x300a,0x03a6);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcmm::DataElement &tblocksq = nestedds.GetDataElement( tblocksq );
//std::cout << "blocksq" << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sssqi = tblocksq.GetValueAsSQ();
const gdcmm::Item &item3 = sssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds3 = item3.GetNestedDataSet();

gdcmm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcmm::DataElement &tblockdata = nestedds3.GetDataElement( tblockdata );
// std::cout << "blockdata" << std::endl;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( tblockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcmm::Attribute<0x300a,0x0104> bnpts; // IS [179 ] # 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcmm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( blocknpts );
//std::cout << "bnpts.GetValue() " << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
    float x[3] = {};
    x[0] = (float)ptr[2*i+0];
    x[1] = (float)ptr[2*i+1];
    //x[2] = ptr[i+2];
    vtkIdType ptId = newPts->InsertNextPoint( x );

```

```

        //std::cout << x[0] << "," << x[1] << "," << x[2] << std::endl;
        ptIds[i] = ptId;
    }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
    #if (VTK_MAJOR_VERSION >= 6)
    #else
        output->Update();
    #endif
    output->Print( std::cout );

// }

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

    vtkImageColorViewer *viewer = vtkImageColorViewer::New();
    #if (VTK_MAJOR_VERSION >= 6)
        viewer->SetInputData(img);
    #else
        viewer->SetInput(img);
    #endif
    viewer->SetupInteractor(iren);
    viewer->SetSize(600, 600);
    viewer->GetRenderer()->ResetCameraClippingRange();
    viewer->Render();
    viewer->GetRenderer()->ResetCameraClippingRange();

    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputData( output );
    #else
        cubeMapper->SetInput( output );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty *property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    viewer->GetRenderer()->AddActor( cubeActor );

    vtkXMLPolyDataWriter *writec= vtkXMLPolyDataWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writec->SetInputData( output );
    #else
        writec->SetInput( output );
    #endif
    writec->SetFileName( outfilename2 );
    writec->Write( );

    iren->Initialize();
    iren->Start();

    return 0;
}

```

14.160 gdcmrtpplan.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

```

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
This example is just for fun. We found a fake RT Ion Plan Storage and simply extracted the viz stuff for VTK
but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];

    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
        return 1;
    }

    /*
(300a,00b0) SQ                                # u/1,1 Beam Sequence
(fffe,e000) na (Item with undefined length)
(300a,00b2) SH (no value)                      # 0,1 Treatment Machine Name
(300a,00c0) IS [1 ]                            # 2,1 Beam Number
(300a,00c2) LO [1 ]                            # 2,1 Beam Name
(300a,00c4) CS [STATIC]                        # 6,1 Beam Type
(300a,00c6) CS [PROTON]                        # 6,1 Radiation Type
(300a,00ce) CS [TREATMENT ]                   # 10,1 Treatment Delivery Type
(300a,00e0) IS [1 ]                            # 2,1 Number of Compensators
(300a,00e3) SQ                                # u/1,1 Compensator Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [lucite]                        # 6,1 Material ID
(300a,00e4) IS [1 ]                            # 2,1 Compensator Number
(300a,00e5) SH [75hdhe5 ]                     # 8,1 Compensator ID
(300a,00e7) IS [35]                           # 2,1 Compensator Rows
(300a,00e8) IS [37]                           # 2,1 Compensator Columns
(300a,00e9) DS [3.679991\4.249288 ]            # 18,2 Compensator Pixel Spacing
(300a,00ea) DS [-76.00\62.50]                  # 12,2 Compensator Position
(300a,00ec) DS
[52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.88\44.88]
# 7618,1-n Compensator Thickness Data
(300a,02e0) CS [ABSENT]                        # 6,1 Compensator Divergence
(300a,02e1) CS [SOURCE_SIDE ]                 # 12,1 Compensator Mounting Position
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)

```

```

    (ffe,e0dd)
*/
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
gdcm::Tag tbeamsq(0x300a,0x00b0);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcm::DataElement &tbeamsq = ds.GetDataElement( tbeamsq );
//std::cout << "beamsq" << std::endl;
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = tbeamsq.GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcm::Item & item = sqi->GetItem(2); // Item start at #1
const gdcm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << "nestedds" << std::endl;
gdcm::Tag tcompensatorsq(0x300a,0x00e3);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcm::DataElement &tcompensatorsq = nestedds.GetDataElement( tcompensatorsq );
//std::cout << "compensatorsq" << std::endl;
gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = tcompensatorsq.GetValueAsSQ();
const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << "nestedds2" << std::endl;
gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
{
    return 1;
}
const gdcm::DataElement &tcompensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
// std::cout << "compensatorthicknessdata" << std::endl;
gdcm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( tcompensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1 Compensator Rows
gdcm::Attribute<0x300a,0x00e7> at1;
const gdcm::DataElement &tcompensatorrows = nestedds2.GetDataElement( at1.GetTag() );
at1.SetFromDataElement( tcompensatorrows );
std::cout << "at1.GetValue()" << std::endl;
// (300a,00e8) IS [37] # 2,1 Compensator Columns
gdcm::Attribute<0x300a,0x00e8> at2;
const gdcm::DataElement &tcompensatorcols = nestedds2.GetDataElement( at2.GetTag() );
at2.SetFromDataElement( tcompensatorcols );
std::cout << "at2.GetValue()" << std::endl;

// (300a,00e9) DS [3.679991\4.249288 ] # 18,2 Compensator Pixel Spacing
gdcm::Attribute<0x300a,0x00e9> at3;
const gdcm::DataElement &tcompensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
at3.SetFromDataElement( tcompensatorpixelspacing );
std::cout << "at3.GetValue(0)" << std::endl;
// (300a,00ea) DS [-76.00\62.50] # 12,2 Compensator Position
gdcm::Attribute<0x300a,0x00ea> at4;
const gdcm::DataElement &tcompensatorposition = nestedds2.GetDataElement( at4.GetTag() );
at4.SetFromDataElement( tcompensatorposition );
std::cout << "at4.GetValue(0)" << std::endl;

vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

vtkImageData *img = vtkImageData::New();
img->Initialize();
img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
//imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);

```



```

#else
img->SetNumberOfScalarComponents(1);
#endif
img->GetPointData()->SetScalars(d);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
(300a,00f4) SQ # u/1,1 Block Sequence
    (ffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ] # 6,1 Material ID
        (300a,00f8) CS [APERTURE] # 8,1 Block Type
        (300a,00fa) CS [ABSENT] # 6,1 Block Divergence
        (300a,00fb) CS [SOURCE_SIDE ] # 12,1 Block Mounting Position
        (300a,00fc) IS [1 ] # 2,1 Block Number
        (300a,0100) DS [50.00 ] # 6,1 Block Thickness
        (300a,0104) IS [179 ] # 4,1 Block Number of Points
        (300a,0106) DS
            [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2\44.6\28.9\44.2\29.7\43.7]
            # 1934,2-2n Block Data
    (ffe,e00d)
    (ffe,e000) na (Item with undefined length)
    (ffe,e00d)
    (ffe,e0dd)
*/
gdcmm::Tag tblocksq(0x300a,0x00f4);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcmm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << "blocksq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
const gdcmm::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds3 = item3.GetNestedDataSet();

gdcmm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcmm::DataElement &tblockdata = nestedds3.GetDataElement( tblockdata );
// std::cout << "blockdata << std::endl;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( blockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcmm::Attribute<0x300a,0x0104> bnpts; // IS [179 ] # 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcmm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
bnpts.SetFromDataElement( blocknpts );
std::cout << bnpts.GetValue() << std::endl;

vtkPolyData *output = vtkPolyData::New();
vtkPoints *newPts = vtkPoints::New();
vtkCellArray *polys = vtkCellArray::New();
const double *ptr = at_.GetValues();
//unsigned int npts = bnpts.GetNumberOfValues() / 2;
unsigned int npts = bnpts.GetValue();
vtkIdType *ptIds = new vtkIdType[npts];
for(unsigned int i = 0; i < npts; ++i)
{
    float x[3] = {};
    x[0] = (float)ptr[2*i+0];
    x[1] = (float)ptr[2*i+1];
    //x[2] = pts[i+2];
    vtkIdType ptId = newPts->InsertNextPoint( x );
    //std::cout << x[0] << ", " << x[1] << ", " << x[2] << std::endl;
    ptIds[i] = ptId;
}

```

```

    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
    output->Print( std::cout );

// }

    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

    vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
    viewer->SetInputData(img);
#else
    viewer->SetInput(img);
#endif
    viewer->SetupInteractor(iren);
    viewer->SetSize(600, 600);
    viewer->Render();

    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
#if (VTK_MAJOR_VERSION >= 6)
    cubeMapper->SetInputData( output );
#else
    cubeMapper->SetInput( output );
#endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty *property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();

    viewer->GetRenderer()->AddActor( cubeActor );

    iren->Initialize();
    iren->Start();

    return 0;
}

```

14.161 gdcmscene.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMPolyDataReader.h"
// #include "vtkGDCMPolyDataWriter.h"

#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"

```

```

#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkVersion.h"

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " filename1.dcm\n";
        return 1;
    }
    const char * filename = argv[1];

    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( filename );
    reader->Update();

    // vtkGDCMPolyDataWriter * writer2 = vtkGDCMPolyDataWriter::New();
    // for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
    //     writer2->SetInput( num, reader->GetOutput(num) );
    // writer2->SetFileName( "rtstruct.dcm" );
    // writer2->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    vtkAppendPolyData *append = vtkAppendPolyData::New();
    int n = reader->GetNumberOfOutputPorts();
    for(int i = 0; i < n; ++i)
    {
        #if (VTK_MAJOR_VERSION >= 6)
            append->AddInputConnection( reader->GetOutputPort(i) );
        #else
            append->AddInput( reader->GetOutput(i) );
        #endif
    }

    vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( reader->GetOutputPort() );
    #else
        writer->SetInput( reader->GetOutput() );
    #endif
    writer->SetFileName( "rtstruct.vtk" );
    //writer->Write();

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
    //cubeMapper->SetInput( reader->GetOutput() );
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputConnection( append->GetOutputPort() );
    #else
        cubeMapper->SetInput( append->GetOutput() );
    #endif
    cubeMapper->SetScalarRange(0,7);
    vtkActor *cubeActor = vtkActor::New();
    //vtkActor2D* cubeActor = vtkActor2D::New();
    cubeActor->SetMapper(cubeMapper);
    vtkProperty * property = cubeActor->GetProperty();
    property->SetRepresentationToWireframe();
    //cubeActor->GetProperty()->SetColor(1, 0, 0);

    // The usual rendering stuff.
    // vtkCamera *camera = vtkCamera::New();
    // camera->SetPosition(1,1,1);

```

```
// camera->SetFocalPoint(0,0,0);

vtkRenderer *renderer = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(renderer);

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renWin);

renderer->AddActor(cubeActor);
//renderer->AddActor2D(cubeActor);
//renderer->SetActiveCamera(camera);
renderer->ResetCamera();
renderer->SetBackground(1,1,1);

renWin->SetSize(300,300);

// interact with data
renWin->Render();
iren->Start();

reader->Delete();
append->Delete();
cubeMapper->Delete();
cubeActor->Delete();
// camera->Delete();
renderer->Delete();
renWin->Delete();
iren->Delete();

writer->Delete();

return 0;
}
```

14.162 gdcmttexture.cxx

```
/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
```

```

reader->SetFileName( argv[1] );

reader->Update();
vtkImageData* ima = reader->GetOutput();

vtkLookupTable* table = vtkLookupTable::New();
table->SetNumberOfColors(1000);
table->SetTableRange(0,1000);
table->SetSaturationRange(0,0);
table->SetHueRange(0,1);
table->SetValueRange(0,1);
table->SetAlphaRange(1,1);
table->Build();

// Texture
vtkTexture* texture = vtkTexture::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
texture->SetInputData(ima);
#else
texture->SetInput(ima);
#endif
texture->InterpolateOn();
texture->SetLookupTable(table);

// PlaneSource
vtkPlaneSource* plane = vtkPlaneSource::New();
plane->SetOrigin( -0.5, -0.5, 0.0);
plane->SetPoint1( 0.5, -0.5, 0.0);
plane->SetPoint2( -0.5, 0.5, 0.0);

// PolyDataMapper
vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
planeMapper->SetInputConnection(plane->GetOutputPort());
#else
planeMapper->SetInput(plane->GetOutput());
#endif

// Actor
vtkActor* planeActor = vtkActor::New();
planeActor->SetTexture(texture);
planeActor->SetMapper(planeMapper);
planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText( "L" );
cube->SetXMinusFaceText( "R" );
cube->SetYPlusFaceText( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText( "H" );
cube->SetZMinusFaceText( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();
// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(180);
reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(reader->GetDirectionCosines());
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
//cube->SetUserTransform( transform ); // can't get it to work
cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
//widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );

```

```

widget->SetInteractor( iren );
//widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

renwin->Render();
iren->Start();

// Clean up:
reader->Delete();
table->Delete();
texture->Delete();
plane->Delete();
planeMapper->Delete();
planeActor->Delete();
ren->Delete();
renwin->Delete();
iren->Delete();

return 0;
}

```

14.163 gdcmvolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdc.m.sourcforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkVersion.h"
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#if VTK_MAJOR_VERSION < 7
#include "vtkVolumeTextureMapper3D.h"
#endif
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkVersion.h"

// gdcmvolume gdcmData/GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( argv[1] );
    reader->Update();

    // Create the renderers, render window, and interactor
    vtkRenderWindow *renWin = vtkRenderWindow::New();
    vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);
    vtkRenderer *ren = vtkRenderer::New();
    renWin->AddRenderer(ren);

    // Create a transfer function mapping scalar value to opacity
    vtkPiecewiseFunction *oTFun = vtkPiecewiseFunction::New();
    //oTFun->AddSegment(0, 1.0, 256, 0.1);
    oTFun->AddSegment(0, 1.0, 240, 0.1);

    vtkColorTransferFunction *cTFun = vtkColorTransferFunction::New();

```

```

cTFun->AddRGBPoint( 0, 1.0, 1.0, 1.0 );
//cTFun->AddRGBPoint( 255, 1.0, 1.0, 1.0 );
cTFun->AddRGBPoint( 240, 1.0, 1.0, 1.0 );

// Need to crop to actually see minimum intensity
vtkImageClip *clip = vtkImageClip::New();
clip->SetInputConnection( reader->GetOutputPort() );
clip->SetOutputWholeExtent(0,66,0,66,30,37);
clip->ClipDataOn();

vtkVolumeProperty *property = vtkVolumeProperty::New();
property->SetScalarOpacity(oTFun);
property->SetColor(cTFun);
property->SetInterpolationTypeToLinear();

vtkFixedPointVolumeRayCastMapper *mapper = vtkFixedPointVolumeRayCastMapper::New();
mapper->SetBlendModeToMinimumIntensity();
mapper->SetInputConnection( reader->GetOutputPort() );

vtkVolume *volume = vtkVolume::New();
volume->SetMapper(mapper);
volume->SetProperty(property);

ren->AddViewProp(volume);

renWin->Render();
{
    iren->Start();
}

volume->Delete();
mapper->Delete();
property->Delete();
clip->Delete();
cTFun->Delete();
oTFun->Delete();
reader->Delete();
renWin->Delete();
iren->Delete();
ren->Delete();

return 0;
}

```

14.164 offscreenimage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkImageMapToWindowLevelColors.h"
#include "vtkImageActor.h"
#include "vtkPNGWriter.h"
#include "vtkWindowToImageFilter.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
}

```

```

const char *filename = argv[1];

vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
reader->SetFileName( filename );
reader->Update(); // important to read the window/level info

vtkMedicalImageProperties *prop = reader->GetMedicalImageProperties();

vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->OffScreenRenderingOn();

vtkRenderer *renderer = vtkRenderer::New();
renWin->AddRenderer(renderer);

vtkImageMapToWindowLevelColors *windowlevel = vtkImageMapToWindowLevelColors::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
windowlevel->SetInputConnection( reader->GetOutputPort() );
#else
windowlevel->SetInput( reader->GetOutput() );
#endif
unsigned int n = prop->GetNumberOfWindowLevelPresets();
if( n )
{
    // Take the first one by default:
    const double *wl = prop->GetNthWindowLevelPreset(0);
    windowlevel->SetWindow( wl[0] );
    windowlevel->SetLevel( wl[1] );
}

vtkImageActor *actor = vtkImageActor::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
actor->SetInputData( windowlevel->GetOutput() );
#else
actor->SetInput( windowlevel->GetOutput() );
#endif

renderer->AddActor( actor );

renWin->Render();

vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
w2if->SetInput ( renWin );

vtkPNGWriter *wr = vtkPNGWriter::New();
#ifdef (VTK_MAJOR_VERSION >= 6)
wr->SetInputConnection( w2if->GetOutputPort() );
#else
wr->SetInput( w2if->GetOutput() );
#endif
#ifdef (VTK_MAJOR_VERSION >= 6)
wr->SetFileName( "offscreenimage.png" );
wr->Write();

reader->Delete();
renWin->Delete();
renderer->Delete();
windowlevel->Delete();
actor->Delete();
w2if->Delete();
wr->Delete();

return 0;
}

```

14.165 reslicesphere.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```



```

===== */
//
// Load a DICOM series.
// Position a sphere within the volume.
// Allow the user to change between Axial, Sagittal, Coronal, and
// Oblique view of the images and move through the slices.
// The display should show the resliced image and the cross section
// of the sphere intersecting that plane.
//

/*
from Scott Johnson /Scott Johnson neuwave com/
to VTK /vtkusers vtk.org/
date Tue, May 11, 2010 at 7:01 PM
*/
#include <string>

#include <vtkDICOMImageReader.h>
#include <vtkStringArray.h>
#include <vtkDirectory.h>
#include <vtkImageThreshold.h>
#include <vtkImageShiftScale.h>
#include <vtkImageReslice.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageViewer2.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkPlane.h>
#include <vtkCutter.h>
#include <vtkActor.h>
#include <vtkCommand.h>
#include <vtkSmartPointer.h>
#include <vtkMatrix4x4.h>
#include <vtkInteractorObserver.h>
#include <vtkProperty.h>
#include <vtkRenderer.h>
#include <vtkImageData.h>
#include <vtkImageActor.h>
#include "vtkTransformPolyDataFilter.h"
#include <vtkCamera.h>
#include <vtkMath.h>
#include <vtkTransform.h>
#include <vtkTextActor.h>
#include <vtkActor2D.h>
#include <vtkPolyDataMapper2D.h>
#include <vtkProperty2D.h>
#include <vtkGDCMImageReader.h>
#include <vtkImageChangeInformation.h>
#include <vtkVersion.h>

#include "gdcmdirctory.h"
#include "gdcmtesting.h"
#include "gdcmiPPSorter.h"

// Change to match the path to find Raw_0.vti or provide
// the parameter when starting ResliceSphere.

const double sphereCenter[3]={74, 219, 70};

// Angles (0, 0, 0)
const double AxialMatrix[] = { 1.0, 0.0, 0.0, 0.0,
                               0.0, 1.0, 0.0, 0.0,
                               0.0, 0.0, 1.0, 0.0,
                               0.0, 0.0, 0.0, 1.0 };

// Angles (0, 90, 0)
const double SagittalMatrix[] = { 0.0, 0.0, 1.0, 0.0,
                                   0.0, 1.0, 0.0, 0.0,
                                   -1.0, 0.0, 0.0, 0.0,
                                   0.0, 0.0, 0.0, 1.0 };

// Angles (-90, 0, 0)
const double CoronalMatrix[] = { 1.0, 0.0, 0.0, 0.0,
                                  0.0, 0.0, 1.0, 0.0,
                                  0.0, -1.0, 0.0, 0.0,
                                  0.0, 0.0, 0.0, 1.0 };

// Angles (0, 90, 31)
const double ObliqueMatrix[] = { 0.0, -0.515038, 0.857167, 0.0,
                                   0.0, 0.857167, 0.515038, 0.0,
                                   -1.0, 0.0, 0.0, 0.0,

```

```

        0.0, 0.0,    0.0,    1.0 };

class ResliceRender;

// Class to handle key press events.
class KeyCallback : public vtkCommand
{
public:
    static KeyCallback* New()
    {
        return new KeyCallback();
    }

    void Execute(vtkObject* caller, unsigned long eventId, void *calldata);
    void SetCallbackData(ResliceRender* reslice);

protected:
    ResliceRender* _reslice;
};

class ResliceRender
{
public:
    typedef enum __ORIENTATION
    {
        AXIAL = 0,
        SAGITTAL = 1,
        CORONAL = 2,
        OBLIQUE = 3
    } ORIENTATION;

    ResliceRender()
    {
        _orientation=AXIAL;
    }

    ~ResliceRender()
    {
        _transform->Delete();
        _reader->Delete();
        _reslice->Delete();
        _interactor->Delete();
        _imageViewer->Delete();

        _sphere->Delete();
        _sphereMapper->Delete();
        _sphereActor->Delete();

        _plane->Delete();
        _cutter->Delete();
        _polyTransform->Delete();
        _ROIMapper->Delete();
        _ROIActor->Delete();

        _annotation->Delete();
    }

    void CreatePipeline(const char* fileName)
    {
        vtkProperty2D* props;

        //_reader=vtkXMLImageDataReader::New();
        //_reader->SetFileName(fileName);
        //_reader->Update();

        //_reader=qzDICOMImageReader::New();
        _reader=vtkGDCMImageReader::New();

        //vtkDirectory *d = vtkDirectory::New();
        //d->Open(fileName);
        //d->Print( std::cout );
        gdcm::Directory d;
        d.Load(fileName);
        gdcm::Directory::FileNamesType const &files = d.GetFileNames();

        gdcm::IPPSorter s;
        s.SetComputeZSpacing( true );
        s.SetZSpacingTolerance( 1e-3 );
        bool b = s.Sort( files );
        if( !b )
        {

```

```

std::cerr << "Failed to sort:" << fileName << std::endl;
//return ;
}
//std::cout << "Sorting succeeded:" << std::endl;
//s.Print( std::cout );

//std::cout << "Found z-spacing:" << std::endl;
//std::cout << s.GetZSpacing() << std::endl;
double ippzspacing = s.GetZSpacing();

const std::vector<std::string> & sorted = s.GetFilesNames();
vtkStringArray *vtkfiles = vtkStringArray::New();
std::vector< std::string >::const_iterator it = sorted.begin();
for( ; it != sorted.end(); ++it)
{
    const std::string &f = *it;
    vtkfiles->InsertNextValue( f.c_str() );
}

//_reader->SetDirectoryName(fileName);
//_reader->SetFileNames( d->GetFiles() );
_reader->SetFileNames( vtkfiles );
_reader->Update();

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();

vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
v16->SetInputConnection( _reader->GetOutputPort() );
#else
v16->SetInput( _reader->GetOutput() );
#endif
v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
v16->Update();

_threshold=vtkImageThreshold::New();
_threshold->ThresholdByUpper(-3024.0);
_threshold->ReplaceOutOn();
_threshold->SetOutValue(0.0);
_threshold->SetInputConnection(v16->GetOutputPort());

_shift=vtkImageShiftScale::New();
_shift->SetShift(0);
_shift->SetScale(1);
_shift->SetInputConnection(_threshold->GetOutputPort());

// Initialize the reslice with an axial orientation.
vtkSmartPointer<vtkMatrix4x4> matrix =
    vtkSmartPointer<vtkMatrix4x4>::New();
matrix->Identity();

_transform = vtkTransform::New();
_transform->SetMatrix(matrix);

_reslice = vtkImageReslice::New();
_reslice->SetOutputDimensionality(3);

// PROBLEM:
// The original intent was to connect the same transform
// to the vtkImageReslice and vtkTransformPolyDataFilter,
// but the resulting reslices appear different using the
// vtkTransform as opposed to explicitly setting the
// reslice axes via SetResliceAxes. Also, if the vtkTransform
// is connected and orientated other than axial, the extents
// don't seem to update resulting in VTK believing the slice
// is out of range.

//_reslice->SetResliceTransform(_transform);
_reslice->SetResliceAxes(matrix);
//_reslice->SetInputConnection(_reader->GetOutputPort());
_reslice->SetInputConnection(_shift->GetOutputPort());

// Create the sphere target shape.
_sphere=vtkSphereSource::New();
_sphere->SetRadius(7.0);
_sphere->SetThetaResolution(16);
_sphere->SetPhiResolution(16);

```

```

__sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

__sphereMapper=vtkPolyDataMapper::New();
__sphereMapper->SetInputConnection(__sphere->GetOutputPort());

__sphereActor=vtkActor::New();
__sphereActor->SetMapper(__sphereMapper);
__sphereActor->PickableOff();
__sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
__sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
__sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
__sphereActor->SetVisibility(true);

// Create the cutting pipeline.
// This plane will be positioned in the original image coordinate system.
__plane = vtkPlane::New();
__plane->SetNormal(0.0, 0.0, 1.0);

__cutter = vtkCutter::New();
__cutter->SetInputConnection(__sphere->GetOutputPort());
__cutter->SetCutFunction(__plane);
__cutter->GenerateCutScalarsOn();
__cutter->SetValue(0, 0.5);

// The transform attached to __polyTransform should move the cut
// ROI into the resliced coordinate system, which should be the
// same as the coordinate system of the resliced images.
// PROBLEM: It doesn't.
__polyTransform = vtkTransformPolyDataFilter::New();
__polyTransform->SetTransform(__transform);
__polyTransform->SetInputConnection(__cutter->GetOutputPort());

__ROIMapper = vtkPolyDataMapper2D::New();
__ROIMapper->SetInputConnection(__polyTransform->GetOutputPort());

vtkCoordinate* coordinate = vtkCoordinate::New();
coordinate->SetCoordinateSystemToWorld();
__ROIMapper->SetTransformCoordinate(coordinate);

__ROIActor = vtkActor2D::New();
__ROIActor->SetMapper(__ROIMapper);

// Make sure the cut can be seen, especially the edges.
props=__ROIActor->GetProperty();
props->SetLineWidth(2);
props->SetOpacity(1.0);
// props->EdgeVisibilityOn();
// props->SetDiffuse(0.8);
// props->SetSpecular(0.3);
// props->SetSpecularPower(20);
// props->SetRepresentationToSurface();
// props->SetDiffuseColor(1.0, 0.0, 0.0);
// props->SetEdgeColor(1.0, 0.0, 0.0);
props->SetColor(1.0, 0.0, 0.0);

__interactor = vtkRenderWindowInteractor::New();

// Create the image viewer and add the actor with the cut ROI.
__imageViewer = vtkImageViewer2::New();
__imageViewer->SetupInteractor(__interactor);
__imageViewer->SetSize(400, 400);
__imageViewer->SetColorWindow(1024);
__imageViewer->SetColorLevel(800);
__imageViewer->SetInputConnection(__reslice->GetOutputPort());
__imageViewer->GetImageActor()->SetOpacity(0.5);

__annotation = vtkTextActor::New();
__annotation->SetTextScaleModeToViewport();
__imageViewer->GetRenderer()->AddActor(__annotation);

// Add the cut shape actor to the renderer.
__imageViewer->GetRenderer()->AddActor(__ROIActor);

// Set up the key handler.
vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
callback->SetCallbackData(this);
__interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

__interactor->Initialize();

```

```

}

void Start()
{
    _interactor->Start();
}

void ResetOrientation()
{
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    SetOrientation(matrix);
}

// Make sure the orientation of the vtkImageReslice and
// vtkTransform are in sync.
void SetOrientation(vtkMatrix4x4* matrix)
{
    _reslice->SetResliceAxes(matrix);
    _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    _transform->SetMatrix(inverse);
    _transform->Update();
}

// Set the current slice of the current view.
void SetSlice(int slice)
{
    std::stringstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM: It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
#ifdef (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    imageData->UpdateInformation();
#endif

    // Let vtkImageViewer2 handle the slice limits.
    _imageView->SetSlice(slice);
    newSlice=GetSlice();

    imageData->GetCenter(center);
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    // Compute the position of the center of the slice based on the
    // spacing of the slices. The resliced axis will always
    // be the "Z" axis.
    point[0]=center[0];
    point[1]=center[1];
    point[2]=(newSlice * spacing[2]) + origin[2];
    point[3]=1.0;

    // Convert the coordinate from the reslice coordinate system to the
    // original image coordinate system.
    // PROBLEM: Logically this seems like it should have been multiplied
    // by the inverse to translate from the resliced coordinate system to
    // the original coordinate system. However, multiplying by the inverse
    // sticks the plane in the wrong place completely. Using the original
    // matrix at least gets the Z coordinate right.
    vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
    vtkSmartPointer<vtkMatrix4x4> inverse =
        vtkSmartPointer<vtkMatrix4x4>::New();

```

```

    vtkMatrix4x4::Invert(matrix, inverse);

    matrix->MultiplyPoint(point, newPoint);
    _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

    // Annotate the image.
    posString « "Position: (" « newPoint[0] « ", " « newPoint[1]
        « ", " « newPoint[2] « ") Slice: " « newSlice;
    _annotation->SetInput(posString.str().c_str());

    _imageView->Render();
}

int GetSlice()
{
    return _imageView->GetSlice();
}

// Set the orientation of the view.
void SetOrientation(ResliceRender::ORIENTATION orientation)
{
    vtkCamera* camera=_imageView->GetRenderer()->GetActiveCamera();

    double spacing[3];
    double origin[3];
    double point[4];
    double newPoint[4];
    double initialPosition;
    double xDirCosine[3];
    double yDirCosine[3];
    double zDirCosine[3];
    double normal[3];

    vtkImageData* imageData;

    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();

    _orientation=orientation;

    // Reset ViewUp
    camera->SetViewUp(0.0, 1.0, 0.0);

    // Compute the cut plane position to the input coordinate system.
    imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
#ifdef (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    imageData->UpdateInformation();
#endif
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    point[0]=origin[0];
    point[1]=origin[1];
    point[2]=origin[2];
    point[3]=1.0;

    switch (_orientation)
    {
    case AXIAL:
        matrix->DeepCopy(AxialMatrix);
        initialPosition=sphereCenter[2];
        break;

    case CORONAL:
        matrix->DeepCopy(CoronalMatrix);
        initialPosition=sphereCenter[1];
        break;

    case SAGITTAL:
        matrix->DeepCopy(SagittalMatrix);
        initialPosition=sphereCenter[0];
        break;

    case OBLIQUE:
        matrix->DeepCopy(ObliqueMatrix);
        initialPosition=sphereCenter[2];
        break;
    }
}

```

```

    }

    // Move the origin from the original image coordinate system to the
    // resliced image coordinate system.
    matrix->MultiplyPoint(point, newPoint);
    matrix->SetElement(0, 3, newPoint[0]);
    matrix->SetElement(1, 3, newPoint[1]);
    matrix->SetElement(2, 3, newPoint[2]);

    ResetOrientation();
    SetOrientation(matrix);

    // Compute the cutting plane normal and set it.
    // PROBLEM: If the transformation is connected rather than
    // using SetResliceAxes, the Direction Cosines do not reflect
    // the orientation of the vtkImageReslice.
    _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                             zDirCosine);
    vtkMath::Cross(xDirCosine, yDirCosine, normal);
    _plane->SetNormal(normal);

    // Set the extents and spacing of the reslice to account for
    // all of the data.
    _reslice->SetOutputExtentToDefault();
    _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);

    // Force the vtkImageViewer2 to update.
    // PROBLEM: The whole extent does not seem to be set in time
    // for the first render. This results in an error because the
    // slice is positioned outside the old bounds.
    #if (VTK_MAJOR_VERSION >= 6)
        _imageView->SetInputData(NULL);
    #else
        _imageView->SetInput(NULL);
    #endif
    _imageView->SetInputConnection(_reslice->GetOutputPort());

    _imageView->GetRenderer()->ResetCameraClippingRange();
    _imageView->GetRenderer()->ResetCamera();

    // Set the initial slice to be at the center of the sphere.
    // Divide by the spacing because this will be undone in SetSlice.
    SetSlice( (int)(initialPosition / spacing[0]));
}

vtkRenderWindowInteractor* GetInteractor()
{
    return _interactor;
}

protected:
    ORIENTATION      _orientation;

    //qzDICOMImageReader*    _reader;
    vtkGDCMImageReader*    _reader;
    vtkImageThreshold*      _threshold;
    vtkImageShiftScale*     _shift;
    vtkImageReslice*        _reslice;
    vtkRenderWindowInteractor* _interactor;
    vtkImageViewer2*        _imageView;

    vtkSphereSource*        _sphere;
    vtkPolyDataMapper*      _sphereMapper;
    vtkActor*               _sphereActor;

    vtkPlane*               _plane;
    vtkCutter*              _cutter;
    vtkTransform*           _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*     _ROIMapper;
    vtkActor2D*             _ROIActor;

    vtkTextActor*           _annotation;
};

// Catch KeyPress events.
// Up Arrow - increases the slice
// Down Arrow - decreases the slice
// 'A' - sets the view to Axial
// 'S' - sets the view to Sagittal

```

```

// 'C'      - sets the view to Coronal
// 'O'      - set the view to Oblique

void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *calldata)
{
    (void)caller;
    (void)eventId;
    (void)calldata;
    std::string sym=__reslice->GetInteractor()->GetKeySym();

    if (!sym.compare("Up"))
    {
        __reslice->SetSlice(__reslice->GetSlice() + 1);
    }
    else if (!sym.compare("Down"))
    {
        __reslice->SetSlice(__reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        __reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        __reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        __reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        __reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}

void KeyCallback::SetCallbackData(ResliceRender* reslice)
{
    __reslice=reslice;
}

// Usage: ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;

    if (argc == 1)
    {
        const char *root = gdcm::Testing::GetDataExtraRoot();
        std::string dir3 = root;
        dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
        render.CreatePipeline(dir3.c_str());
    }
    else
    {
        render.CreatePipeline(argv[1]);
    }

    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();

    return EXIT_SUCCESS;
}

```

14.166 rtstructapp.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====

```



```

===== */
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"

#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

/*
 * Small example to read in a RTSTRUCT and write it out (displays it too).
 */

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];
    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( filename );
    reader->Update();

    //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() << std::endl;

    vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
    writer->SetNumberOfInputPorts( reader->GetNumberOfOutputPorts() );
    writer->SetFileName( outfilename );
    for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( num, reader->GetOutputPort(num) );
    #else
        writer->SetInput( num, reader->GetOutput(num) );
    #endif
    //doesn't look like the medical properties are actually written out
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
    writer->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    vtkAppendPolyData *append = vtkAppendPolyData::New();

    int n = reader->GetNumberOfOutputPorts();
    for(int i = 0; i < n; ++i)
    {
        #if (VTK_MAJOR_VERSION >= 6)
            append->AddInputConnection( reader->GetOutputPort(i) );
        #else
            append->AddInput( reader->GetOutput(i) );
        #endif
    }

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cubeMapper->SetInputConnection( append->GetOutputPort());
    #else
        cubeMapper->SetInput( append->GetOutput());
    #endif
}

```

```

cubeMapper->SetScalarRange(0,7);
vtkActor *cubeActor = vtkActor::New();
cubeActor->SetMapper(cubeMapper);
vtkProperty *property = cubeActor->GetProperty();
property->SetRepresentationToWireframe();

vtkRenderer *renderer = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(renderer);

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renWin);

renderer->AddActor(cubeActor);
renderer->ResetCamera();
renderer->SetBackground(1,1,1);

renWin->SetSize(300,300);

renWin->Render();
iren->Start();

reader->Delete();
append->Delete();
cubeMapper->Delete();
cubeActor->Delete();
renderer->Delete();
renWin->Delete();
iren->Delete();
writer->Delete();

return 0;
}

```

14.167 threadgdcmm.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmImageReader.h"
#include "gdcmmDirectory.h"
#include "gdcmmSystem.h"

#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"
#include "vtkVersion.h"

#include <pthread.h>

struct threadparams
{
    const char **filenames;
    size_t nfiles;
    char *scalarpainter;
    // TODO I should also pass in the dim of the reference image just in case
};

void *ReadFilesThread(void *voidparams)
{
    const threadparams *params = static_cast<const threadparams *> (voidparams);

    const size_t nfiles = params->nfiles;
    for(unsigned int file = 0; file < nfiles; ++file)
    {
        /*
        // TODO: update progress

```

```

pthread_mutex_lock(&params->lock);
//section critique
ReadingProgress+=params->stepProgress;
pthread_mutex_unlock(&params->lock);
*/
const char *filename = params->filenames[file];
//std::cerr « filename « std::endl;

gdcmm::ImageReader reader;
reader.SetFileName( filename );
try
{
    if( !reader.Read() )
    {
        std::cerr « "Failed to read: " « filename « std::endl;
        break;
    }
}
catch( ... )
{
    std::cerr « "Failed to read: " « filename « std::endl;
    break;
}

const gdcmm::Image &image = reader.GetImage();
unsigned long len = image.GetBufferLength();
char * pointer = params->scalarpointer;
#if 0
char *tempimage = new char[len];
image.GetBuffer(tempimage);

memcpy(pointer + file*len, tempimage, len);
delete[] tempimage;
#else
char *tempimage = pointer + file * len;
image.GetBuffer(tempimage);
#endif
}

return voidparams;
}

void ShowFilenames(const threadparams &params)
{
    std::cout « "start" « std::endl;
    for(unsigned int i = 0; i < params.nfiles; ++i)
    {
        const char *filename = params.filenames[i];
        std::cout « filename « std::endl;
    }
    std::cout « "end" « std::endl;
}

void ReadFiles(size_t nfiles, const char *filenames[])
{
    // \precondition: nfiles > 0
    assert( nfiles > 0 );
    const char *reference= filenames[0]; // take the first image as reference

    gdcmm::ImageReader reader;
    reader.SetFileName( reference );
    if( !reader.Read() )
    {
        // That would be very bad...
        assert(0);
    }

    const gdcmm::Image &image = reader.GetImage();
    gdcmm::PixelFormat pixeltype = image.GetPixelFormat();
    unsigned long len = image.GetBufferLength();
    const unsigned int *dims = image.GetDimensions();
    unsigned short pixelsize = pixeltype.GetPixelSize();
    (void)pixelsize;
    assert( image.GetNumberOfDimensions() == 2 );

    vtkImageData *output = vtkImageData::New();
    output->SetDimensions(dims[0], dims[1], (int)nfiles);

    #if (VTK_MAJOR_VERSION >= 6)
    int numscal = pixeltype.GetSamplesPerPixel();
    switch( pixeltype )

```

```

{
case gdcm::PixelFormat::INT8:
output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
break;
case gdcm::PixelFormat::UINT8:
output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
break;
case gdcm::PixelFormat::INT16:
output->AllocateScalars( VTK_SHORT, numscal );
break;
case gdcm::PixelFormat::UINT16:
output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
break;
case gdcm::PixelFormat::INT32:
output->AllocateScalars( VTK_INT, numscal );
break;
case gdcm::PixelFormat::UINT32:
output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
break;
default:
assert(0);
}
#else
switch( pixeltype )
{
case gdcm::PixelFormat::INT8:
#if ( VTK_MAJOR_VERSION >= 5 ) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
output->SetScalarType ( VTK_SIGNED_CHAR );
#else
output->SetScalarType ( VTK_CHAR );
#endif
break;
case gdcm::PixelFormat::UINT8:
output->SetScalarType ( VTK_UNSIGNED_CHAR );
break;
case gdcm::PixelFormat::INT16:
output->SetScalarType ( VTK_SHORT );
break;
case gdcm::PixelFormat::UINT16:
output->SetScalarType ( VTK_UNSIGNED_SHORT );
break;
case gdcm::PixelFormat::INT32:
output->SetScalarType ( VTK_INT );
break;
case gdcm::PixelFormat::UINT32:
output->SetScalarType ( VTK_UNSIGNED_INT );
break;
default:
assert(0);
}
output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );
output->AllocateScalars();
#endif
char * scalarpointer = static_cast<char*>(output->GetScalarPointer());

const unsigned int nthreads = 4;
threadparams params[nthreads];

//pthread_mutex_t lock;
//pthread_mutex_init(&lock, NULL);

pthread_t *pthread = new pthread_t[nthreads];

// There is nfiles, and nThreads
assert( nfiles > nthreads );
const size_t partition = nfiles / nthreads;
for (unsigned int thread=0; thread < nthreads; ++thread)
{
params[thread].filenames = filenames + thread * partition;
params[thread].nfiles = partition;
if( thread == nthreads - 1 )
{
// There is slightly more files to process in this thread:
params[thread].nfiles += nfiles % nthreads;
}
assert( thread * partition < nfiles );
params[thread].scalarpointer = scalarpointer + thread * partition * len;
//assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] * dims[1] * dims[2] );
// start thread:
int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[thread] );
if( res )

```

```

        {
            std::cerr << "Unable to start a new thread, pthread returned: " << res << std::endl;
            assert(0);
        }
        //ShowFileNames(params[thread]);
    }
// DEBUG
    size_t total = 0;
    for (unsigned int thread=0; thread < nthreads; ++thread)
    {
        total += params[thread].nfiles;
    }
    assert( total == nfiles );
// END DEBUG

    for (unsigned int thread=0;thread<nthreads;thread++)
    {
        pthread_join( pthread[thread], NULL);
    }
    delete[] pthread;

    //pthread_mutex_destroy(&lock);

    // For some reason writing down the file is painfully slow...
    vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputData( output );
    #else
        writer->SetInput( output );
    #endif
    writer->SetFileName( "/tmp/threadgdcmm.vtk" );
    writer->SetFileTypeToBinary();
    //writer->Write();
    writer->Delete();

    //output->Print( std::cout );
    output->Delete();
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " [directory|list of filenames]\n";
        return 1;
    }

    // Check if user pass in a single directory
    if( argc == 2 && gdcmm::System::FileIsDirectory( argv[1] ) )
    {
        gdcmm::Directory d;
        d.Load( argv[1] );
        gdcmm::Directory::FileNamesType l = d.GetFilesNames();
        const size_t nfiles = l.size();
        const char **filenames = new const char* [ nfiles ];
        for(unsigned int i = 0; i < nfiles; ++i)
        {
            filenames[i] = l[i].c_str();
        }
        ReadFiles(nfiles, filenames);
        delete[] filenames;
    }
    else
    {
        // Simply copy all filenames into the vector:
        const char **filenames = const_cast<const char**>(argv+1);
        const size_t nfiles = argc - 1;
        ReadFiles(nfiles, filenames);
    }

    return 0;
}

```

14.168 AWTMedical3.java

```

/*=====

```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```
=====*/
package examples;

import vtk.*;
//import gdcm.*;

import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;

import java.util.ArrayList;

import javax.swing.*;
import java.awt.*;
import java.io.File;

public class AWTMedical3 extends JComponent implements VtkPanelContainer {

    private vtkPanel renWin;

    vtkImageData ReadDataFile(File inSelectedFile){

        vtkImageData outImageData = null;
        Directory theDir = new Directory();

        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);

        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag);//get studies,
        theScanner.AddTag(theSeriesTag);//get studies,
        theScanner.Scan(theDir.GetFilesNames());

        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag);
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
        //vector of strings
        if (theNumStudies != 1)
            return outImageData;
        String theStudyVal = theStudyValues.get(0);
        //now, get all the values from the scanner that are in that
        //study, then from that get their different series
        FilenamesType theFilesNames =
            theScanner.GetAllFilesNamesFromTagToValue(theStudyTag, theStudyVal);

        //from that set of filenames, isolate individual series
        //conclude that singleton series = RT struct (can do further
        //checking for things like MIPs and the like)
        //and multiple series entries = volumetric data
        theScanner.Scan(theFilesNames);
        FilenamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
        String studyUID = theScanner.GetValue(theScanner.GetFilesNames().get(0), theStudyTag);
        long theNumSeries = theSeriesValues.size();
        for (int i = 0; i < theNumSeries; i++) {
            FilenamesType theSeriesFiles =
                theScanner.GetAllFilesNamesFromTagToValue(theSeriesTag, theSeriesValues.get(i));
            long theNumFilesInSeries = theSeriesFiles.size();
            if (theNumFilesInSeries > 1) { //assume it's CT or volumetric data
                //for now, assume a single volume
                //could have multiples, like PET and CT

                IPPSorter sorter = new IPPSorter();
                sorter.SetComputeZSpacing(true);
                sorter.SetZSpacingTolerance(0.001);
                Boolean sorted = sorter.Sort(theSeriesFiles);
                if (!sorted){
```

```

        //need some better way to handle failures here
        return outImageData;
    }

    FilenamesType sortedFT = sorter.GetFilenames();
    long theSize = sortedFT.size();
    vtkStringArray sa = new vtkStringArray();
    ArrayList<String> theStrings = new ArrayList<String>();

    vtkGDCMImageReader gdcmReader = new vtkGDCMImageReader();
    for (int j = 0; j < theSize; j++) {
        String theFileName = sortedFT.get(j);
        if (gdcmReader.CanReadFile(theFileName) > 0){
            theStrings.add(theFileName);
            sa.InsertNextValue(theFileName);
        } else {
            //this is a busted series
            //need some more appropriate error here
            return outImageData;
        }
    }

    gdcmReader.SetFileNames(sa);

    gdcmReader.Update();

    outImageData = gdcmReader.GetOutput();//the zeroth output should be the image
}
String theImageInfo = "";
if (outImageData != null){
    theImageInfo = outImageData.Print();
}
return outImageData;
}

//this function is a rewrite of Medical3 to see if data can
//be loaded via gdcm easily
public AWTMedical3(File inFile) {
    // Create the buttons.
    renWin = new vtkPanel();

    vtkImageData theImageData = ReadDataFile(inFile);

    // An isosurface, or contour value of 500 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);
    skinExtractor.SetValue(0, 500);
    vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
    skinNormals.SetInput(skinExtractor.GetOutput());
    skinNormals.SetFeatureAngle(60.0);
    // vtkStripper skinStripper = new vtkStripper();
    // skinStripper.SetInput(skinNormals.GetOutput());
    vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
    skinMapper.SetInput(skinNormals.GetOutput());
    skinMapper.ScalarVisibilityOff();
    vtkActor skin = new vtkActor();
    skin.SetMapper(skinMapper);
    skin.GetProperty().SetDiffuseColor(1, .49, .25);
    skin.GetProperty().SetSpecular(.3);
    skin.GetProperty().SetSpecularPower(20);

    // An isosurface, or contour value of 1150 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter boneExtractor = new vtkContourFilter();
    boneExtractor.SetInput(theImageData);
    boneExtractor.SetValue(0, 1150);
    vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
    boneNormals.SetInput(boneExtractor.GetOutput());
    boneNormals.SetFeatureAngle(60.0);
    vtkStripper boneStripper = new vtkStripper();
    boneStripper.SetInput(boneNormals.GetOutput());
    vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
    boneMapper.SetInput(boneStripper.GetOutput());

```

```

boneMapper.ScalarVisibilityOff();
vtkActor bone = new vtkActor();
bone.SetMapper(boneMapper);
bone.GetProperty().SetDiffuseColor(1, 1, .9412);

// An outline provides context around the data.
vtkOutlineFilter outlineData = new vtkOutlineFilter();
outlineData.SetInput(theImageData);
vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
mapOutline.SetInput(outlineData.GetOutput());
vtkActor outline = new vtkActor();
outline.SetMapper(mapOutline);
outline.GetProperty().SetColor(0, 0, 0);

// Now we are creating three orthogonal planes passing through the
// volume. Each plane uses a different texture map and therefore has
// different coloration.

// Start by creating a black/white lookup table.
vtkLookupTable bwLut = new vtkLookupTable();
bwLut.SetTableRange(0, 2000);
bwLut.SetSaturationRange(0, 0);
bwLut.SetHueRange(0, 0);
bwLut.SetValueRange(0, 1);
bwLut.Build();

// Now create a lookup table that consists of the full hue circle (from
// HSV);
vtkLookupTable hueLut = new vtkLookupTable();
hueLut.SetTableRange(0, 2000);
hueLut.SetHueRange(0, 1);
hueLut.SetSaturationRange(1, 1);
hueLut.SetValueRange(1, 1);
hueLut.Build();

// Finally, create a lookup table with a single hue but having a range
// in the saturation of the hue.
vtkLookupTable satLut = new vtkLookupTable();
satLut.SetTableRange(0, 2000);
satLut.SetHueRange(.6, .6);
satLut.SetSaturationRange(0, 1);
satLut.SetValueRange(1, 1);
satLut.Build();

// Create the first of the three planes. The filter vtkImageMapToColors
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane. It does this using texture
// mapping and as a result is quite fast. (Note: the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline
// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors sagittalColors = new vtkImageMapToColors();
sagittalColors.SetInput(theImageData);
sagittalColors.SetLookupTable(bwLut);
vtkImageActor sagittal = new vtkImageActor();
sagittal.SetInput(sagittalColors.GetOutput());
sagittal.SetDisplayExtent(32, 32, 0, 63, 0, 92);

// Create the second (axial); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors axialColors = new vtkImageMapToColors();
axialColors.SetInput(theImageData);
axialColors.SetLookupTable(hueLut);
vtkImageActor axial = new vtkImageActor();
axial.SetInput(axialColors.GetOutput());
axial.SetDisplayExtent(0, 63, 0, 63, 46, 46);

// Create the third (coronal); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors coronalColors = new vtkImageMapToColors();
coronalColors.SetInput(theImageData);
coronalColors.SetLookupTable(satLut);
vtkImageActor coronal = new vtkImageActor();
coronal.SetInput(coronalColors.GetOutput());
coronal.SetDisplayExtent(0, 63, 32, 32, 0, 92);

// It is convenient to create an initial view of the data. The FocalPoint
// and Position form a vector direction. Later on (ResetCamera() method)
// this vector is used to position the camera to look at the data in

```



```

// this direction.
vtkCamera aCamera = new vtkCamera();
aCamera.SetViewUp(0, 0, -1);
aCamera.SetPosition(0, 1, 0);
aCamera.SetFocalPoint(0, 0, 0);
aCamera.ComputeViewPlaneNormal();

// Actors are added to the renderer. An initial camera view is created.
// The Dolly() method moves the camera towards the FocalPoint,
// thereby enlarging the image.
renWin.GetRenderer().AddActor(sagittal);
renWin.GetRenderer().AddActor(axial);
renWin.GetRenderer().AddActor(coronal);
renWin.GetRenderer().AddActor(outline);
renWin.GetRenderer().AddActor(skin);
renWin.GetRenderer().AddActor(bone);

// Turn off bone for this example.
bone.VisibilityOff();

// Set skin to semi-transparent.
skin.GetProperty().SetOpacity(0.5);

// An initial camera view is created. The Dolly() method moves
// the camera towards the FocalPoint, thereby enlarging the image.
renWin.GetRenderer().SetActiveCamera(aCamera);
renWin.GetRenderer().ResetCamera();
aCamera.Dolly(1.5);

// Set a background color for the renderer and set the size of the
// render window (expressed in pixels).
renWin.GetRenderer().SetBackground(1, 1, 1);
VtkPanelUtil.setSize(renWin, 640, 480);

// Note that when camera movement occurs (as it does in the Dolly()
// method), the clipping planes often need adjusting. Clipping planes
// consist of two planes: near and far along the view direction. The
// near plane clips out objects in front of the plane the far plane
// clips out objects behind the plane. This way only what is drawn
// between the planes is actually rendered.
renWin.GetRenderer().ResetCameraClippingRange();

// Setup panel
setLayout(new BorderLayout());
add(renWin, BorderLayout.CENTER);
}

public vtkPanel getRenWin() {
    return renWin;
}

public static void main(String s[]) {
    if (s.length == 0){
        return; //need a filename here
    }
    File theFile = new File(s[0]);
    //File theFile = new
        File("/Users/mmrorden/Documents/MVSDownloadDirectory/Documents/1.2.840.113704.1.111.3384.1271766367.5/");
    AWTMedical3 panel = new AWTMedical3(theFile);

    JFrame frame = new JFrame("AWTMedical3");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);
}
}

```

14.169 HelloVTKWorld.java

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre

All rights reserved.

See Copyright.txt or <http://gdcn.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcn.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdcn.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcn.jar:gdcn.jar:. java HelloVTKWorld gdcnData/012345.002.050.dcm bla.dcm
 */
public class HelloVTKWorld
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcnJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }

    public static void main(String[] args)
    {
        String filename = args[0];
        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileName( filename );
        reader.Update();

        vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
        System.out.println( prop.GetPatientName() ); //

        // if( reader.GetImageFormat() == vtkgdcn.vtkgdcn.VTK_LUMINANCE ) // MONOCHROME2
        // {
        //     System.out.println( "Image is MONOCHROME2" ); //
        // }

        // Just for fun, invert the direction cosines, output should reflect that:
        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        // We need to maintain in sync information stored in vtkMedicalImageProperties:
        double[] cosines = new double[6];
        cosines[0] = dircos.GetElement(0,0);
        cosines[1] = dircos.GetElement(1,0);
        cosines[2] = dircos.GetElement(2,0);
        cosines[3] = dircos.GetElement(0,1);
        cosines[4] = dircos.GetElement(1,1);
        cosines[5] = dircos.GetElement(2,1);
        reader.GetMedicalImageProperties().SetDirectionCosine( cosines );

        String outfilename = args[1];
        vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
    }
}

```

```

        writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
        writer.SetDirectionCosines( dircos );
        writer.SetShift( reader.GetShift() );
        writer.SetScale( reader.GetScale() );
        writer.SetImageFormat( reader.GetImageFormat() );
        writer.SetFileName( outfilename );
        writer.SetInputConnection( reader.GetOutputPort() ); // new
        //writer.SetInput( reader.GetOutput() ); // old
        writer.Write();

        System.out.println("Success reading: " + filename );
    }
}

```

14.170 MIPViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdc.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdc.*;
import java.io.File;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdc.jar:/usr/share/java/vtk.jar javac MIPViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdc.jar:gdc.jar:. java MIPViewer BRAINX
 *
 */
public class MIPViewer extends Canvas
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
        System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
        // VTK-GDCM
        System.loadLibrary("vtkgdc.jar");
    }

    static FilenamesType fns = new FilenamesType();

    protected native int Lock();

    protected native int UnLock();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)

```

```

        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
else
    {
        process(dir.getPath());
    }
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // Create our volume and mapper
    vtkVolume volume = new vtkVolume();
    vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();

    // Add a box widget if the clip option was selected
    vtkBoxWidget box = new vtkBoxWidget();
    box.SetInteractor(iren);
    box.SetPlaceFactor(1.01);
    box.SetInputConnection(change.GetOutputPort());

    //box.SetDefaultRenderer(renderer);
    box.InsideOutOn();
    box.PlaceWidget();
    //vtkBoxWidgetCallback callback = vtkBoxWidgetCallback::New();
    //callback.SetMapper(mapper);
    //box.AddObserver(vtkCommand::InteractionEvent, callback);
    //callback.Delete();
    // Lock();
    // box.EnabledOn();
    // Unlock();
    box.GetSelectedFaceProperty().SetOpacity(0.0);

    mapper.SetInputConnection( change.GetOutputPort() );

    // Create our transfer function
    vtkColorTransferFunction colorFun = new vtkColorTransferFunction();
    vtkPiecewiseFunction opacityFun = new vtkPiecewiseFunction();

    // Create the property and attach the transfer functions
    vtkVolumeProperty property = new vtkVolumeProperty();

```

```

property.IndependentComponentsOn();
property.SetColor( colorFun );
property.SetScalarOpacity( opacityFun );
property.SetInterpolationTypeToLinear();

// connect up the volume to the property and the mapper
volume.SetProperty( property );
volume.SetMapper( mapper );

vtkMedicalImageProperties medprop = reader.GetMedicalImageProperties();
int n = medprop.GetNumberOfWindowLevelPresets();
double opacityWindow = 4096;
double opacityLevel = 2048;

// Override default with value from DICOM files:
for( int i = 0; i < n; ++i )
{
    double wl[] = medprop.GetNthWindowLevelPreset(i);
    //System.out.println( "W/L: " + wl[0] + " " + wl[1] );
    opacityWindow = wl[0];
    opacityLevel = wl[1];
}

colorFun.AddRGBSegment(0.0, 1.0, 1.0, 1.0, 255.0, 1.0, 1.0, 1.0 );
opacityFun.AddSegment( opacityLevel - 0.5*opacityWindow, 0.0,
    opacityLevel + 0.5*opacityWindow, 1.0 );
mapper.SetBlendModeToMaximumIntensity();

// Create the RenderWindow, Renderer
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);

// Set the default window size
renWin.SetSize(600,600);

// Add the volume to the scene
ren1.AddVolume( volume );
ren1.ResetCamera();

iren.SetRenderWindow( renWin );

// interact with data
renWin.Render();

iren.Start();
}
}

```

14.171 MPRViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer BRAINX
 */

```

```

public class MPRViewer
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public static void main(String[] args) throws Exception
    {
        String dirname = args[0];
        if( !PosixEmulation.FileIsDirectory( dirname ) )
        {
            return;
        }

        File dir = new File(dirname);
        visitAllFiles(dir);

        IPPSorter ipp = new IPPSorter();
        ipp.SetComputeZSpacing( true );
        ipp.SetZSpacingTolerance( 1e-3 );
        boolean b = ipp.Sort( fns );
        if(!b)
        {
            throw new Exception("Could not scan");
        }
        double ippzspacing = ipp.GetZSpacing();

        FilenamesType sorted = ipp.GetFilenames();
        vtkStringArray files = new vtkStringArray();
        long nfiles = sorted.size();
        //for( String f : sorted )
        for (int i = 0; i < nfiles; i++) {
            String f = sorted.get(i);
            files.InsertNextValue( f );
        }
        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileNames( files );
        reader.Update(); // get spacing value

        double[] spacing = reader.GetOutput().GetSpacing();

        vtkImageChangeInformation change = new vtkImageChangeInformation();
        change.SetInputConnection( reader.GetOutputPort() );
        change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

        // A simple vtkInteractorStyleImage example for
        // 3D image viewing with the vtkImageResliceMapper.
        //
        // Drag Left mouse button to window/level
    }
}

```

```

// Shift-Left drag to rotate (oblique slice)
// Shift-Middle drag to slice through image
// OR Ctrl-Right drag to slice through image

// Create the RenderWindow, Renderer
vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);

vtkImageResliceMapper im = new vtkImageResliceMapper();
im.SetInputConnection(change.GetOutputPort());
im.SliceFacesCameraOn();
im.SliceAtFocalPointOn();
im.BorderOff();

vtkImageProperty ip = new vtkImageProperty();
ip.SetColorWindow(2000);
ip.SetColorLevel(1000);
ip.SetAmbient(0.0);
ip.SetDiffuse(1.0);
ip.SetOpacity(1.0);
ip.SetInterpolationTypeToLinear();

vtkImageSlice ia = new vtkImageSlice();
ia.SetMapper(im);
ia.SetProperty(ip);

ren1.AddViewProp(ia);
ren1.SetBackground(0.1,0.2,0.4);
renWin.SetSize(300,300);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
vtkInteractorStyleImage style = new vtkInteractorStyleImage();
style.SetInteractionModeToImage3D();
iren.SetInteractorStyle(style);
renWin.SetInteractor(iren);

// render the image
renWin.Render();
vtkCamera cam1 = ren1.GetActiveCamera();
cam1.ParallelProjectionOn();
ren1.ResetCameraClippingRange();
renWin.Render();

iren.Start();
}
}

```

14.172 MPRViewer2.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcms.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcms.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
 * CLASSPATH=/usr/share/java/vtk.jar:vtkgdcms.jar:gdcms.jar:. java MPRViewer2 BRAINX
 */
public class MPRViewer2

```

```

{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkRenderingJava");
        System.loadLibrary("vtkHybridJava");
        System.loadLibrary("vtkWidgetsJava");
        // VTK-GDCM
        System.loadLibrary("vtkgdcmJava");
    }

    static FilenamesType fns = new FilenamesType();

    public static void process(String path)
    {
        fns.add( path );
    }

    // Process only files under dir
    public static void visitAllFiles(File dir)
    {
        if (dir.isDirectory())
        {
            String[] children = dir.list();
            for (int i=0; i<children.length; i++)
            {
                visitAllFiles(new File(dir, children[i]));
            }
        }
        else
        {
            process(dir.getPath());
        }
    }

    public void dointer(vtkImagePlaneWidget current_widget)
    {
        int cstat = current_widget.GetCursorDataStatus();
        double[] v = current_widget.GetCurrentCursorPosition();
        //System.out.println( cstat );
        //System.out.println( v[0] );
        //System.out.println( v[1] );
        //System.out.println( v[2] );
        planeWidgetX.SetSliceIndex( (int)v[0] );
        planeWidgetY.SetSliceIndex( (int)v[1] );
        planeWidgetZ.SetSliceIndex( (int)v[2] );
        planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
        planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }

    public void startinterX()
    {
        dointer( planeWidgetX );
    }

    public void interX()
    {
        dointer( planeWidgetX );
    }

    public void endinterX()
    {
    }

    public void startinterY()
    {
        dointer( planeWidgetY );
    }

    public void interY()
    {
        dointer( planeWidgetY );
    }

    public void endinterY()
    {
    }

    public void startinterZ()
    {
        dointer( planeWidgetZ );
    }

    public void interZ()

```



```

    {
        dointer( planeWidgetZ );
    }
    public void endinterZ()
    {
        //System.out.println( "endinter" );
    }

    public static void AlignCamera(int slice_number, vtkImagePlaneWidget current_widget)
    {
        vtkImageData image = (vtkImageData)current_widget.GetInput();
        vtkRenderer ren = current_widget.GetCurrentRenderer();
        double[] origin = image.GetOrigin();
        double ox = origin[0];
        double oy = origin[1];
        double oz = origin[2];

        int dims[] = image.GetDimensions();
        int xMin = 0;
        int xMax = 1;
        int yMin = 2;
        int yMax = dims[0]-1;
        int zMin = dims[1]-1;
        int zMax = dims[2]-1;

        double[] spacing = image.GetSpacing();
        double sx = spacing[0];
        double sy = spacing[1];
        double sz = spacing[2];

        double cx = ox+(0.5*(xMax-xMin))*sx;
        double cy = oy+(0.5*(yMax-yMin))*sy;
        double cz = oz+(0.5*(zMax-zMin))*sz;
        double vx = 0, vy = 0, vz = 0;
        double nx = 0, ny = 0, nz = 0;
        int iaxis = current_widget.GetPlaneOrientation();
        if ( iaxis == 0 ) {
            vz = -1;
            nx = ox + xMax*sx;
            cx = ox + slice_number*sx;
        }
        else if ( iaxis == 1 ) {
            vz = -1;
            ny = oy+yMax*sy;
            cy = oy+slice_number*sy;
        }
        else {
            vy = 1;
            nz = oz+zMax*sz;
            cz = oz+slice_number*sz;
        }
        double px = cx+nx*2;
        double py = cy+ny*2;
        double pz = cz+nz*3;

        vtkCamera camera = ren.GetActiveCamera();
        camera.SetViewUp(vx, vy, vz);
        camera.SetFocalPoint(cx, cy, cz);
        camera.SetPosition(px, py, pz);
        camera.OrthogonalizeViewUp();
        ren.ResetCameraClippingRange();
    }

    private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
    private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
    private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();

    public void config()
    {
        //System.out.println( "config" );
        planeWidgetX.GetCurrentRenderer().ResetCamera();
        planeWidgetY.GetCurrentRenderer().ResetCamera();
        planeWidgetZ.GetCurrentRenderer().ResetCamera();
    }

    public void Run(String dirname)
    {
        File dir = new File(dirname);
        visitAllFiles(dir);

        IPPSorter ipp = new IPPSorter();

```

```

ipp.SetComputeZSpacing( true );
ipp.SetZSpacingTolerance( 1e-3 );
boolean b = ipp.Sort( fns );
if(!b)
{
    //throw new Exception("Could not scan");
}
double ippzspacing = ipp.GetZSpacing();

FileNamesType sorted = ipp.GetFileNames();
vtkStringArray files = new vtkStringArray();
long nfiles = sorted.size();
//for( String f : sorted )
for (int i = 0; i < nfiles; i++) {
    String f = sorted.get(i);
    files.InsertNextValue( f );
}
vtkGDCMImageReader reader = new vtkGDCMImageReader();
reader.SetFileNames( files );
reader.Update(); // get spacing value

double[] spacing = reader.GetOutput().GetSpacing();

vtkImageChangeInformation change = new vtkImageChangeInformation();
change.SetInputConnection( reader.GetOutputPort() );
change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
change.Update();

System.out.println( change.GetOutput().toString() );

vtkRenderer ren1 = new vtkRenderer();
ren1.SetViewport(0., 0., 0.333, 1);
ren1.SetBackground(0.1,0.2,0.4);
vtkRenderer ren2 = new vtkRenderer();
ren2.SetViewport(0.333, 0., 0.667, 1);
ren2.SetBackground(0.1,0.2,0.4);
vtkRenderer ren3 = new vtkRenderer();
ren3.SetViewport(0.667, 0., 1., 1.);
ren3.SetBackground(0.1,0.2,0.4);

vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);
renWin.AddRenderer(ren2);
renWin.AddRenderer(ren3);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
iren.SetRenderWindow(renWin);

vtkInteractorStyleImage style = new vtkInteractorStyleImage();
iren.SetInteractorStyle( style );

vtkCellPicker picker = new vtkCellPicker();
picker.SetTolerance(0.005);

vtkProperty ipwProp = new vtkProperty();

//vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
planeWidgetX.SetInteractor(iren);
planeWidgetX.SetCurrentRenderer(ren1);
planeWidgetX.SetDefaultRenderer(ren1);
planeWidgetX.RestrictPlaneToVolumeOn();
planeWidgetX.SetTexturePlaneProperty(ipwProp);
//planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetX.TextureInterpolateOff();
//planeWidgetX.SetResliceInterpolateToNearestNeighbour();
planeWidgetX.SetInputConnection(change.GetOutputPort());
planeWidgetX.SetPlaneOrientationToXAxes();
planeWidgetX.SetSliceIndex(62);
planeWidgetX.SetPicker(picker);
planeWidgetX.SetKeyPressActivationValue('x');
planeWidgetX.On();
planeWidgetX.InteractionOn();

//vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
planeWidgetY.SetInteractor(iren);
planeWidgetY.SetCurrentRenderer(ren2);
planeWidgetY.SetDefaultRenderer(ren2);
planeWidgetY.RestrictPlaneToVolumeOn();
planeWidgetY.SetTexturePlaneProperty(ipwProp);
//planeWidgetY.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetY.TextureInterpolateOff();

```

```

//planeWidgetY.SetResliceInterpolateToNearestNeighbour();
planeWidgetY.SetInputConnection(change.GetOutputPort());
planeWidgetY.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetY.SetPlaneOrientationToYAxes();
planeWidgetY.SetSliceIndex(32);
planeWidgetY.SetPicker(picker);
planeWidgetY.SetKeyPressActivationValue('y');
planeWidgetY.On();

//vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
planeWidgetZ.SetInteractor(iren);
planeWidgetZ.SetCurrentRenderer(ren3);
planeWidgetZ.SetDefaultRenderer(ren3);
planeWidgetZ.RestrictPlaneToVolumeOn();
planeWidgetZ.SetTexturePlaneProperty(ipwProp);
//planeWidgetZ.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetZ.TextureInterpolateOff();
//planeWidgetZ.SetResliceInterpolateToNearestNeighbour();
planeWidgetZ.SetInputConnection(change.GetOutputPort());
planeWidgetZ.SetLookupTable( planeWidgetX.GetLookupTable() );
planeWidgetZ.SetPlaneOrientationToZAxes();
planeWidgetZ.SetSliceIndex(32);
planeWidgetZ.SetPicker(picker);
planeWidgetZ.SetKeyPressActivationValue('z');
planeWidgetZ.On();

iren.Initialize();

renWin.Render();
AlignCamera(52, planeWidgetX);
AlignCamera(32, planeWidgetY);
AlignCamera(32, planeWidgetZ);

planeWidgetX.GetCurrentRenderer().ResetCamera();
planeWidgetY.GetCurrentRenderer().ResetCamera();
planeWidgetZ.GetCurrentRenderer().ResetCamera();

renWin.Render();

planeWidgetX.AddObserver("StartInteractionEvent", this,"startinterX");
planeWidgetX.AddObserver("InteractionEvent", this,"interX");
planeWidgetX.AddObserver("EndInteractionEvent", this,"endinterX");
planeWidgetY.AddObserver("StartInteractionEvent", this,"startinterY");
planeWidgetY.AddObserver("InteractionEvent", this,"interY");
planeWidgetY.AddObserver("EndInteractionEvent", this,"endinterY");
planeWidgetZ.AddObserver("StartInteractionEvent", this,"startinterZ");
planeWidgetZ.AddObserver("InteractionEvent", this,"interZ");
planeWidgetZ.AddObserver("EndInteractionEvent", this,"endinterZ");

iren.AddObserver("ConfigureEvent", this,"config");

iren.Start();
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    MPRViewer2 me = new MPRViewer2();
    me.Run( dirname );
}
}

```

14.173 ReadSeriesIntoVTK.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

===== */
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdc.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdc.jar:/usr/share/java/vtk.jar:. ReadSeriesIntoVTK
 */
public class ReadSeriesIntoVTK
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }

    public static void main(String[] args)
    {
        vtkFileOutputWindow outWin = new vtkFileOutputWindow();
        outWin.SetInstance(outWin);
        outWin.SetFileName("MVSvtkViewer.log");

        // See: http://review.source.kitware.com/#change,888
        // vtkWrapJava does not handle static keyword
        // String directory = vtkGDCMTesting.GetGDCMDataRoot();
        vtkGDCMTesting t = new vtkGDCMTesting();
        String directory = t.GetGDCMDataRoot();
        String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
        String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
        String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
        String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";

        vtkStringArray s = new vtkStringArray();
        System.out.println("adding : " + file0 );
        s.InsertNextValue( file0 );
        s.InsertNextValue( file1 );
        s.InsertNextValue( file2 );
        s.InsertNextValue( file3 );

        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileNames( s );
        reader.Update();

        System.out.println("Success reading: " + file0 );

        vtkMetaImageWriter writer = new vtkMetaImageWriter();
        writer.DebugOn();
        writer.SetCompression( false );
        writer.SetInputConnection( reader.GetOutputPort() );
        writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
        writer.Write();

        System.out.println("Success writing: " + writer.GetFileName() );
    }
}

```

14.174 CastConvertPhilips.py

```

00001
00014
00015 """
00016 Usage:
00017
00018 python --public /path/to/directory/
00019 or
00020 python --private /path/to/directory/
00021
00022 python --public --extension bak /path/to/directory/
00023
00024 rename -f 's/\.bak$/ /' *.bak
00025
00026 TODO:
00027 http://docs.python.org/library/optparse.html#module-optparse
00028 """
00029
00030 import vtkgdc
00031 import vtk
00032 import sys
00033 import gdc
00034
00035 def ProcessOneFilePublic(filename, outfilename, tmpfile):
00036     gdc.ImageHelper.SetForceRescaleInterceptSlope(True)
00037     vtkreader = vtkgdc.vtkGDCMImageReader()
00038     vtkreader.SetFileName( filename )
00039     vtkreader.Update()
00040
00041     cast = vtk.vtkImageCast()
00042     cast.SetInput( vtkreader.GetOutput() )
00043     cast.SetOutputScalarTypeToUnsignedShort()
00044
00045     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
00046     # Some operation will actually be discarded (we simply need a temp storage)
00047     vtkwriter = vtkgdc.vtkGDCMImageWriter()
00048     vtkwriter.SetFileName( tmpfile )
00049     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
00050     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
00051     print "Format:",vtkreader.GetImageFormat()
00052     vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
00053     vtkwriter.SetInput( cast.GetOutput() )
00054     #vtkwriter.Update()
00055     vtkwriter.Write()
00056
00057     # ok now rewrite the exact same file as the original (keep all info)
00058     # but use the Pixel Data Element from the written file
00059     tmpreader = gdc.ImageReader()
00060     tmpreader.SetFileName( tmpfile )
00061     if not tmpreader.Read():
00062         sys.exit(1)
00063
00064     reader = gdc.Reader()
00065     reader.SetFileName( filename )
00066     if not reader.Read():
00067         sys.exit(1)
00068
00069     # Make sure to remove Slope/Rescale to avoid re-execution
00070     ds = reader.GetFile().GetDataSet()
00071     tags = [
00072         gdc.Tag(0x0028,0x1052),
00073         gdc.Tag(0x0028,0x1053),
00074         gdc.Tag(0x0028,0x1053),
00075     ]
00076     for tag in tags:
00077         ds.Remove( tag )
00078
00079     writer = gdc.ImageWriter()
00080     writer.SetFileName( outfilename )
00081     # Pass image from vtk written file
00082     writer.SetImage( tmpreader.GetImage() )
00083     # pass dataset from initial 'reader'
00084     writer.SetFile( reader.GetFile() )
00085     if not writer.Write():
00086         sys.exit(1)
00087
00088 def ProcessOneFilePrivate(filename, outfilename, tmpfile):
00089     vtkreader = vtkgdc.vtkGDCMImageReader()
00090     vtkreader.SetFileName( filename )

```

```

00091 vtkreader.Update()
00092
00093
00094 # (2005,1409)   DS    4    0.0
00095 # (2005,140a)   DS   16   1.52283272283272
00096
00097 # (2005,0014)   LO   26   Philips MR Imaging DD 005
00098 tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
00099 tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
00100
00101
00102
00103 # Need to access some private tags, reread the file (for now):
00104 reader = gdcm.Reader()
00105 reader.SetFileName( filename )
00106 if not reader.Read():
00107     sys.exit(1)
00108
00109 ds = reader.GetFile().GetDataSet()
00110
00111 el1 = ds.GetDataElement( tag1 )
00112 el2 = ds.GetDataElement( tag2 )
00113
00114
00115 #pf = gdcm.PythonFilter()
00116 #pf.SetFile( reader.GetFile() )
00117 #print el1.GetTag()
00118
00119 print el1.GetByteValue()
00120 v1 = eval(el1.GetByteValue().GetBuffer())
00121 print el2.GetByteValue()
00122 v2 = eval(el2.GetByteValue().GetBuffer())
00123
00124 print v1
00125 shift = v1
00126 print v2
00127 scale = v2
00128
00129 ss = vtk.vtkImageShiftScale()
00130 ss.SetInput( vtkreader.GetOutput() )
00131 # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
00132 assert shift == 0
00133 ss.SetShift( shift )
00134 ss.SetScale( scale )
00135 ss.SetOutputScalarTypeToUnsignedShort ()
00136 ss.Update()
00137
00138 # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
00139 # Some operation will actually be discarded (we simply need a temp storage)
00140 vtkwriter = vtkgdcml.vtkGDCMImageWriter()
00141 vtkwriter.SetFileName( tmpfile )
00142 vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
00143 vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
00144 vtkwriter.SetImageFormat( reader.GetImageFormat() )
00145 # do not pass shift/scale again
00146 vtkwriter.SetInput( ss.GetOutput() )
00147 #vtkwriter.Update()
00148 vtkwriter.Write()
00149
00150 # ok now rewrite the exact same file as the original (keep all info)
00151 # but use the Pixel Data Element from the written file
00152 tmpreader = gdcm.ImageReader()
00153 tmpreader.SetFileName( tmpfile )
00154 if not tmpreader.Read():
00155     sys.exit(1)
00156
00157 writer = gdcm.ImageWriter()
00158 writer.SetFileName( outfilename )
00159 # Pass image from vtk written file
00160 writer.SetImage( tmpreader.GetImage() )
00161 # pass dataset from initial 'reader'
00162 writer.SetFile( reader.GetFile() )
00163 if not writer.Write():
00164     sys.exit(1)
00165
00166 if __name__ == "__main__":
00167     gdcm.Trace.DebugOff()
00168     gdcm.Trace.WarningOff()
00169     #filename = sys.argv[1]
00170     #outfilename = sys.argv[2]

```

```

00172 tmpfile = "/tmp/philips_rescaled.dcm"
00173 #ProcessOneFile( filename, outfilename, tmpfile )
00174 rescaletype = sys.argv[1]
00175 assert rescaletype == "--public" or rescaletype == "--private"
00176 dirname = sys.argv[2]
00177 d = gdcm.Directory()
00178 d.Load( dirname )
00179
00180 for f in d.GetFilenames():
00181     #print f
00182     ProcessOneFilePublic( f, f + ".bak", tmpfile )
00183
00184
00185 print "success"

```

14.175 headsq2dcm.py

```

00001
00014
00015 """
00016 Usage:
00017 python headsq2dcm.py -D /path/to/VTKData
00018 """
00019
00020 import vtk
00021 import vtkgdcm
00022 from vtk.util.misc import vtkGetDataRoot
00023 VTK_DATA_ROOT = vtkGetDataRoot()
00024
00025 reader = vtk.vtkVolume16Reader()
00026 reader.SetDataDimensions(64, 64)
00027 reader.SetDataByteOrderToLittleEndian()
00028 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
00029 reader.SetImageRange(1, 93)
00030 reader.SetDataSpacing(3.2, 3.2, 1.5)
00031
00032 cast = vtk.vtkImageCast()
00033 cast.SetInput( reader.GetOutput() )
00034 cast.SetOutputScalarTypeToUnsignedChar()
00035
00036 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
00037 writer = vtkgdcm.vtkGDCMImageWriter()
00038 writer.SetFileName( "headsq.dcm" )
00039 writer.SetInput( reader.GetOutput() )
00040 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
00041 #writer.SetInput( cast.GetOutput() )
00042 writer.SetFileDimensionality( 3 )
00043 writer.Write()

```


Index

- ~ASN1
 - gdcmm::ASN1, [156](#)
- ~AnonymizeEvent
 - gdcmm::AnonymizeEvent, [133](#)
- ~Anonymizer
 - gdcmm::Anonymizer, [138](#)
- ~Attribute
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [189](#)
- ~AudioCodec
 - gdcmm::AudioCodec, [221](#)
- ~BaseCompositeMessage
 - gdcmm::network::BaseCompositeMessage, [226](#)
- ~BaseNormalizedMessage
 - gdcmm::network::BaseNormalizedMessage, [228](#)
- ~BasePDU
 - gdcmm::network::BasePDU, [230](#)
- ~BaseQuery
 - gdcmm::BaseQuery, [233](#)
- ~BaseRootQuery
 - gdcmm::BaseRootQuery, [238](#)
- ~Bitmap
 - gdcmm::Bitmap, [251](#)
- ~BitmapToBitmapFilter
 - gdcmm::BitmapToBitmapFilter, [264](#)
- ~BoxRegion
 - gdcmm::BoxRegion, [267](#)
- ~ByteSwapFilter
 - gdcmm::ByteSwapFilter, [274](#)
- ~ByteValue
 - gdcmm::ByteValue, [278](#)
- ~CAPICryptographicMessageSyntax
 - gdcmm::CAPICryptographicMessageSyntax, [287](#)
- ~CSAHeader
 - gdcmm::CSAHeader, [351](#)
- ~Cleaner
 - gdcmm::Cleaner, [300](#)
- ~Coder
 - gdcmm::Coder, [311](#)
- ~Command
 - gdcmm::Command, [318](#)
- ~CommandDataSet
 - gdcmm::CommandDataSet, [322](#)
- ~CryptoFactory
 - gdcmm::CryptoFactory, [337](#)
- ~CryptographicMessageSyntax
 - gdcmm::CryptographicMessageSyntax, [338](#)
- ~Curve
 - gdcmm::Curve, [366](#)
- ~DICOMDIRGenerator
 - gdcmm::DICOMDIRGenerator, [417](#)
- ~DPath
 - gdcmm::DPath, [451](#)
- ~DataEvent
 - gdcmm::DataEvent, [386](#)
- ~DataSetEvent
 - gdcmm::DataSetEvent, [402](#)
- ~Decoder
 - gdcmm::Decoder, [405](#)
- ~Defs
 - gdcmm::Defs, [408](#)
- ~DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [414](#)
- ~DictConverter
 - gdcmm::DictConverter, [425](#)
- ~DictPrinter
 - gdcmm::DictPrinter, [434](#)
- ~Dicts
 - gdcmm::Dicts, [436](#)
- ~DirectionCosines
 - gdcmm::DirectionCosines, [442](#)
- ~Directory
 - gdcmm::Directory, [445](#)
- ~Dumper
 - gdcmm::Dumper, [455](#)
- ~Element
 - gdcmm::Element< TVR, VM::VM1_n >, [469](#)
- ~EmptyMaskGenerator
 - gdcmm::EmptyMaskGenerator, [521](#)
- ~Event
 - gdcmm::Event, [533](#)
- ~Exception
 - gdcmm::Exception, [537](#)
- ~File
 - gdcmm::File, [550](#)
- ~FileAnonymizer
 - gdcmm::FileAnonymizer, [555](#)
- ~FileChangeTransferSyntax
 - gdcmm::FileChangeTransferSyntax, [559](#)
- ~FileDecompressLookupTable

- gdcmm::FileDecompressLookupTable, [563](#)
- ~FileDerivation
 - gdcmm::FileDerivation, [565](#)
- ~FileExplicitFilter
 - gdcmm::FileExplicitFilter, [569](#)
- ~FileMetaInformation
 - gdcmm::FileMetaInformation, [575](#)
- ~FileNameEvent
 - gdcmm::FileNameEvent, [586](#)
- ~FileStreamer
 - gdcmm::FileStreamer, [596](#)
- ~FilenameGenerator
 - gdcmm::FilenameGenerator, [589](#)
- ~Global
 - gdcmm::Global, [616](#)
- ~GroupDict
 - gdcmm::GroupDict, [619](#)
- ~IconImageFilter
 - gdcmm::IconImageFilter, [622](#)
- ~IconImageGenerator
 - gdcmm::IconImageGenerator, [625](#)
- ~Image
 - gdcmm::Image, [633](#)
- ~ImageApplyLookupTable
 - gdcmm::ImageApplyLookupTable, [639](#)
- ~ImageChangePhotometricInterpretation
 - gdcmm::ImageChangePhotometricInterpretation, [643](#)
- ~ImageChangePlanarConfiguration
 - gdcmm::ImageChangePlanarConfiguration, [648](#)
- ~ImageChangeTransferSyntax
 - gdcmm::ImageChangeTransferSyntax, [653](#)
- ~ImageCodec
 - gdcmm::ImageCodec, [659](#)
- ~ImageConverter
 - gdcmm::ImageConverter, [669](#)
- ~ImageFragmentSplitter
 - gdcmm::ImageFragmentSplitter, [672](#)
- ~ImageReader
 - gdcmm::ImageReader, [683](#)
- ~ImageRegionReader
 - gdcmm::ImageRegionReader, [688](#)
- ~ImageToImageFilter
 - gdcmm::ImageToImageFilter, [692](#)
- ~ImageWriter
 - gdcmm::ImageWriter, [696](#)
- ~JPEG12Codec
 - gdcmm::JPEG12Codec, [731](#)
- ~JPEG16Codec
 - gdcmm::JPEG16Codec, [736](#)
- ~JPEG2000Codec
 - gdcmm::JPEG2000Codec, [741](#)
- ~JPEG8Codec
 - gdcmm::JPEG8Codec, [750](#)
- ~JPEGCodec
 - gdcmm::JPEGCodec, [755](#)
- ~JPEGLSCodec
 - gdcmm::JPEGLSCodec, [763](#)
- ~JSON
 - gdcmm::JSON, [768](#)
- ~KAKADUCodec
 - gdcmm::KAKADUCodec, [772](#)
- ~LookupTable
 - gdcmm::LookupTable, [779](#)
- ~MemberCommand
 - gdcmm::MemberCommand< T >, [809](#)
- ~MeshPrimitive
 - gdcmm::MeshPrimitive, [815](#)
- ~ModuleEntry
 - gdcmm::ModuleEntry, [831](#)
- ~MrProtocol
 - gdcmm::MrProtocol, [844](#)
- ~Object
 - gdcmm::Object, [873](#)
- ~OpenSSLCryptographicMessageSyntax
 - gdcmm::OpenSSLCryptographicMessageSyntax, [879](#)
- ~OpenSSL7CryptographicMessageSyntax
 - gdcmm::OpenSSL7CryptographicMessageSyntax, [884](#)
- ~Orientation
 - gdcmm::Orientation, [887](#)
- ~Overlay
 - gdcmm::Overlay, [892](#)
- ~PDBHeader
 - gdcmm::PDBHeader, [912](#)
- ~PDFCodec
 - gdcmm::PDFCodec, [915](#)
- ~PGXCodec
 - gdcmm::PGXCodec, [925](#)
- ~PNMCodec
 - gdcmm::PNMCodec, [961](#)
- ~PVRGCodec
 - gdcmm::PVRGCodec, [1002](#)
- ~ParseException
 - gdcmm::ParseException, [899](#)
- ~Parser
 - gdcmm::Parser, [902](#)
- ~Pixmap
 - gdcmm::Pixmap, [943](#)
- ~PixmapReader
 - gdcmm::PixmapReader, [949](#)
- ~PixmapToPixmapFilter
 - gdcmm::PixmapToPixmapFilter, [952](#)
- ~PixmapWriter
 - gdcmm::PixmapWriter, [956](#)
- ~Preamble
 - gdcmm::Preamble, [964](#)

- ~Printer
 - gdcm::Printer, [985](#)
- ~PrivateDict
 - gdcm::PrivateDict, [988](#)
- ~ProgressEvent
 - gdcm::ProgressEvent, [997](#)
- ~PythonFilter
 - gdcm::PythonFilter, [1004](#)
- ~QueryBase
 - gdcm::QueryBase, [1006](#)
- ~RAWCodec
 - gdcm::RAWCodec, [1022](#)
- ~RLECodec
 - gdcm::RLECodec, [1043](#)
- ~Reader
 - gdcm::Reader, [1027](#)
- ~Region
 - gdcm::Region, [1034](#)
- ~Rescaler
 - gdcm::Rescaler, [1037](#)
- ~SHA1
 - gdcm::SHA1, [1125](#)
- ~Scanner
 - gdcm::Scanner, [1053](#)
- ~Scanner2
 - gdcm::Scanner2, [1063](#)
- ~Segment
 - gdcm::Segment, [1072](#)
- ~SegmentReader
 - gdcm::SegmentReader, [1085](#)
- ~SegmentWriter
 - gdcm::SegmentWriter, [1090](#)
- ~SegmentedPaletteColorLookupTable
 - gdcm::SegmentedPaletteColorLookupTable, [1081](#)
- ~SerieHelper
 - gdcm::SerieHelper, [1111](#)
- ~ServiceClassUser
 - gdcm::ServiceClassUser, [1119](#)
- ~SimpleMemberCommand
 - gdcm::SimpleMemberCommand< T >, [1129](#)
- ~SimpleSubjectWatcher
 - gdcm::SimpleSubjectWatcher, [1132](#)
- ~SmartPointer
 - gdcm::SmartPointer< ObjectType >, [1139](#)
- ~Sorter
 - gdcm::Sorter, [1145](#)
- ~Spacing
 - gdcm::Spacing, [1150](#)
- ~SplitMosaicFilter
 - gdcm::SplitMosaicFilter, [1152](#)
- ~StreamImageReader
 - gdcm::StreamImageReader, [1159](#)
- ~StreamImageWriter
 - gdcm::StreamImageWriter, [1164](#)
- ~StrictScanner
 - gdcm::StrictScanner, [1173](#)
- ~StrictScanner2
 - gdcm::StrictScanner2, [1183](#)
- ~StringFilter
 - gdcm::StringFilter, [1194](#)
- ~Subject
 - gdcm::Subject, [1199](#)
- ~Surface
 - gdcm::Surface, [1205](#)
- ~SurfaceReader
 - gdcm::SurfaceReader, [1220](#)
- ~SurfaceWriter
 - gdcm::SurfaceWriter, [1224](#)
- ~Table
 - gdcm::Table, [1238](#)
- ~TableEntry
 - gdcm::TableEntry, [1240](#)
- ~TableReader
 - gdcm::TableReader, [1241](#)
- ~TableRow
 - gdcm::network::TableRow, [1244](#)
- ~TagPath
 - gdcm::TagPath, [1256](#)
- ~Testing
 - gdcm::Testing, [1259](#)
- ~Trace
 - gdcm::Trace, [1265](#)
- ~Transition
 - gdcm::network::Transition, [1278](#)
- ~ULAction
 - gdcm::network::ULAction, [1331](#)
- ~ULBasicCallback
 - gdcm::network::ULBasicCallback, [1372](#)
- ~ULConnection
 - gdcm::network::ULConnection, [1374](#)
- ~ULConnectionCallback
 - gdcm::network::ULConnectionCallback, [1379](#)
- ~ULConnectionManager
 - gdcm::network::ULConnectionManager, [1385](#)
- ~ULEvent
 - gdcm::network::ULEvent, [1390](#)
- ~ULWritingCallback
 - gdcm::network::ULWritingCallback, [1394](#)
- ~UserInformation
 - gdcm::network::UserInformation, [1409](#)
- ~Validate
 - gdcm::Validate, [1412](#)
- ~Value
 - gdcm::Value, [1415](#)
- ~Version
 - gdcm::Version, [1419](#)
- ~Writer

- gdcmm::Writer, [1561](#)
- ~XMLDictReader
 - gdcmm::XMLDictReader, [1566](#)
- ~XMLPrinter
 - gdcmm::XMLPrinter, [1568](#)
- ~XMLPrivateDictReader
 - gdcmm::XMLPrivateDictReader, [1572](#)
- ~vtkGDCMImageReader
 - vtkGDCMImageReader, [1448](#)
- ~vtkGDCMImageReader2
 - vtkGDCMImageReader2, [1462](#)
- ~vtkGDCMImageWriter
 - vtkGDCMImageWriter, [1476](#)
- ~vtkGDCMMedicalImageProperties
 - vtkGDCMMedicalImageProperties, [1483](#)
- ~vtkGDCMPolyDataReader
 - vtkGDCMPolyDataReader, [1487](#)
- ~vtkGDCMPolyDataWriter
 - vtkGDCMPolyDataWriter, [1492](#)
- ~vtkGDCMTesting
 - vtkGDCMTesting, [1496](#)
- ~vtkGDCMThreadedImageReader
 - vtkGDCMThreadedImageReader, [1501](#)
- ~vtkGDCMThreadedImageReader2
 - vtkGDCMThreadedImageReader2, [1505](#)
- ~vtkImageColorViewer
 - vtkImageColorViewer, [1513](#)
- ~vtkImageMapToColors16
 - vtkImageMapToColors16, [1524](#)
- ~vtkImageMapToWindowLevelColors2
 - vtkImageMapToWindowLevelColors2, [1530](#)
- ~vtkImagePlanarComponentsToComponents
 - vtkImagePlanarComponentsToComponents, [1534](#)
- ~vtkImageRGBToYBR
 - vtkImageRGBToYBR, [1536](#)
- ~vtkImageYBRToRGB
 - vtkImageYBRToRGB, [1538](#)
- ~vtkLookupTable16
 - vtkLookupTable16, [1541](#)
- ~vtkRTStructSetProperties
 - vtkRTStructSetProperties, [1545](#)
- AAabortPDU
 - gdcmm::network::AAabortPDU, [114](#)
- AAAssociateACPDU
 - gdcmm::network::AAAssociateACPDU, [117](#)
 - gdcmm::network::AAAssociateRQPDU, [127](#)
- AAAssociateRJPDU
 - gdcmm::network::AAAssociateRJPDU, [121](#)
- AAAssociateRQPDU
 - gdcmm::network::AAAssociateACPDU, [119](#)
 - gdcmm::network::AAAssociateRQPDU, [124](#)
- AbstractMultiDimensionalImageModel
 - gdcmm::UIDs, [1311](#)
- AbstractSyntax
 - gdcmm::network::AbstractSyntax, [129](#)
 - gdcmm::PresentationContext, [970](#)
- AcquisitionContextSRStorage
 - gdcmm::UIDs, [1310](#)
- ActiveComponent
 - vtkImageMapToColors16, [1528](#)
- Add
 - gdcmm::GroupDict, [620](#)
- add1
 - gdcmm, [91](#)
- AddAcceptedPresentationContext
 - gdcmm::network::ULConnection, [1374](#)
- AddContourReferencedFrameOfReference
 - vtkRTStructSetProperties, [1545](#)
- AddCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, [356](#)
- AddDerivationDescription
 - gdcmm::FileDerivation, [566](#)
- AddDictEntry
 - gdcmm::Dict, [421](#)
 - gdcmm::PrivateDict, [988](#)
- AddFile
 - gdcmm::FileSet, [592](#)
 - gdcmm::SerieHelper, [1111](#)
- AddFileName
 - gdcmm::SerieHelper, [1111](#)
- AddFragment
 - gdcmm::SequenceOfFragments, [1095](#)
- AddFromFile
 - gdcmm::PresentationContextGenerator, [974](#)
- AddGroupLength
 - gdcmm::DictConverter, [425](#)
- AddImageDirectoryRecord
 - gdcmm::DICOMDIRGenerator, [417](#)
- AddInput
 - vtkImageColorViewer, [1514](#)
- AddInputConnection
 - vtkImageColorViewer, [1514](#)
- AddIOD
 - gdcmm::IODs, [713](#)
- AddIODEntry
 - gdcmm::IOD, [708](#)
- AddItem
 - gdcmm::SequenceOfItems, [1103](#)
- AddMacro
 - gdcmm::Macros, [790](#)
 - gdcmm::Module, [827](#)
- AddMacroEntry
 - gdcmm::Macro, [788](#)
- AddModule
 - gdcmm::Modules, [835](#)
- AddModuleEntry

- gdcmm::Module, [827](#)
- gdcmm::NestedModuleEntries, [857](#)
- AddNewUndefinedLengthItem
 - gdcmm::SequenceOfItems, [1103](#)
- AddObserver
 - gdcmm::Subject, [1200](#)
- AddPatientDirectoryRecord
 - gdcmm::DICOMDIRGenerator, [417](#)
- AddPresentationContext
 - gdcmm::network::AAssociateRQPDU, [124](#)
 - gdcmm::PresentationContextGenerator, [974](#)
- AddPresentationContextAC
 - gdcmm::network::AAssociateACPDU, [118](#)
- AddPresentationDataValue
 - gdcmm::network::PDataTFPDU, [906](#)
- AddPrimitiveData
 - gdcmm::MeshPrimitive, [815](#)
- AddPrivateTag
 - gdcmm::Scanner, [1053](#)
 - gdcmm::Scanner2, [1063](#)
 - gdcmm::StrictScanner, [1173](#)
 - gdcmm::StrictScanner2, [1183](#)
- AddPublicTag
 - gdcmm::Scanner2, [1063](#)
 - gdcmm::StrictScanner2, [1183](#)
- AddPurposeOfReferenceCodeSequence
 - gdcmm::FileDerivation, [566](#)
- AddQueryDataSet
 - gdcmm::BaseQuery, [233](#)
- AddReference
 - gdcmm::FileDerivation, [566](#)
- AddReferencedFrameOfReference
 - vtkRTStructSetProperties, [1546](#)
- AddRestriction
 - gdcmm::SerieHelper, [1111](#)
- AddRoleSelectionSub
 - gdcmm::network::UserInformation, [1409](#)
- AddSegment
 - gdcmm::SegmentWriter, [1090](#)
- AddSelect
 - gdcmm::Sorter, [1146](#)
- AddSeriesDirectoryRecord
 - gdcmm::DICOMDIRGenerator, [418](#)
- AddSkipTag
 - gdcmm::Scanner, [1053](#)
 - gdcmm::Scanner2, [1063](#)
 - gdcmm::StrictScanner, [1173](#)
 - gdcmm::StrictScanner2, [1183](#)
- AddSOPClassExtendedNegociationSub
 - gdcmm::network::UserInformation, [1409](#)
- AddSourceImageSequence
 - gdcmm::FileDerivation, [566](#)
- AddStructureSetROI
 - vtkRTStructSetProperties, [1546](#)
- AddStructureSetROIObservation
 - vtkRTStructSetProperties, [1546](#)
- AddStudyDirectoryRecord
 - gdcmm::DICOMDIRGenerator, [418](#)
- AddSurface
 - gdcmm::Segment, [1072](#)
- AddTag
 - gdcmm::Scanner, [1053](#)
 - gdcmm::StrictScanner, [1174](#)
- AddTransferSyntax
 - gdcmm::network::PresentationContextRQ, [977](#)
 - gdcmm::PresentationContext, [969](#)
- AdultMouseAnatomyOntology
 - gdcmm::UIDs, [1308](#)
- AdvancedBlendingPresentationStateStorage
 - gdcmm::UIDs, [1309](#)
- AE
 - gdcmm::VR, [1432](#)
- AECComp
 - gdcmm, [86](#)
- AES128_CIPHER
 - gdcmm::CryptographicMessageSyntax, [338](#)
- AES192_CIPHER
 - gdcmm::CryptographicMessageSyntax, [338](#)
- AES256_CIPHER
 - gdcmm::CryptographicMessageSyntax, [338](#)
- AffectedSOPClassUID
 - gdcmm::network::CEchoRQ, [291](#)
- AGFA
 - gdcmm::EquipmentManufacturer, [531](#)
- ALGOType
 - gdcmm::Segment, [1072](#)
- ALGOType_END
 - gdcmm::Segment, [1072](#)
- Allocate
 - gdcmm::LookupTable, [780](#)
- AmbulatoryECGWaveformStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- AnatomicRegion
 - gdcmm::Segment, [1076](#)
- AnatomicRegionModifiers
 - gdcmm::Segment, [1076](#)
- AnonymizeEvent
 - gdcmm::AnonymizeEvent, [133](#)
- Anonymizer
 - gdcmm::Anonymizer, [138](#)
- Append
 - gdcmm::ByteValue, [278](#)
 - gdcmm::Global, [616](#)
- AppendFrameEncode
 - gdcmm::ImageCodec, [659](#)
 - gdcmm::JPEG2000Codec, [741](#)
 - gdcmm::JPEGCodec, [755](#)

- gdcmm::JPEGLSCodec, [764](#)
- gdcmm::RLECodec, [1044](#)
- AppendImplementationClassUID
 - gdcmm::FileMetaInformation, [576](#)
- AppendRowEncode
 - gdcmm::ImageCodec, [659](#)
 - gdcmm::JPEG2000Codec, [741](#)
 - gdcmm::JPEGCodec, [755](#)
 - gdcmm::JPEGLSCodec, [764](#)
 - gdcmm::RLECodec, [1044](#)
- AppendToDataElement
 - gdcmm::FileStreamer, [596](#)
- AppendToGroupDataElement
 - gdcmm::FileStreamer, [596](#)
- ApplicationContext
 - gdcmm::network::ApplicationContext, [146](#)
- Apply
 - gdcmm::ImageApplyLookupTable, [640](#)
- ApplyInverseVideo
 - vtkGDCMImageReader, [1456](#)
 - vtkGDCMImageReader2, [1471](#)
- ApplyLookupTable
 - vtkGDCMImageReader, [1456](#)
 - vtkGDCMImageReader2, [1471](#)
- ApplyPlanarConfiguration
 - vtkGDCMImageReader, [1456](#)
 - vtkGDCMImageReader2, [1471](#)
- ApplyShiftScale
 - vtkGDCMImageReader, [1456](#)
 - vtkGDCMImageReader2, [1471](#)
- ApplyYBRToRGB
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1471](#)
- Area
 - gdcmm::BoxRegion, [267](#)
 - gdcmm::Region, [1034](#)
- AResourceRPPDU
 - gdcmm::network::AResourceRPPDU, [150](#)
- AResourceRQPDU
 - gdcmm::network::AResourceRQPDU, [153](#)
- AreOverlaysInPixelData
 - gdcmm::Bitmap, [252](#)
 - gdcmm::Pixmap, [943](#)
- ARGB
 - gdcmm::PhotometricInterpretation, [928](#)
- ArrayIncludeMacrosType
 - gdcmm::Macro, [787](#)
 - gdcmm::Module, [827](#)
- ArrayType
 - gdcmm::Attribute< Group, Element, TVR, TVM
>, [161](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1 >, [170](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_3 >, [178](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_8 >, [183](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_n >, [189](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_2n >, [198](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_n >, [203](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_3n >, [210](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_n >, [216](#)
- ArterialPulseWaveformStorage
 - gdcmm::UIDs, [1309](#)
- ARTIMTimer
 - gdcmm::network::ARTIMTimer, [154](#)
- AS
 - gdcmm::VR, [1432](#)
- ASComp
 - gdcmm, [86](#)
- ASN1
 - gdcmm::ASN1, [156](#)
- AsynchronousOperationsWindowSub
 - gdcmm::network::AsynchronousOperationsWin-
dowSub, [157](#)
- AT
 - gdcmm::VR, [1432](#)
- Attribute
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_n >, [189](#)
 - gdcmm::terminal, [110](#)
- Audio
 - gdcmm::MediaStorage, [802](#)
- AudioCodec
 - gdcmm::AudioCodec, [221](#)
- AudioSRStorageTrialRetired
 - gdcmm::UIDs, [1305](#)
- AUTOMATIC
 - gdcmm::Segment, [1072](#)
- AutoPixelMinMax
 - gdcmm::IconImageGenerator, [625](#)
- AutorefractionMeasurementsStorage
 - gdcmm::UIDs, [1309](#)
- AXIAL
 - gdcmm::Orientation, [887](#)
- backslash
 - gdcmm, [91](#)
- BadBigEndian
 - gdcmm::SwapCode, [1227](#)
- BadLittleEndian
 - gdcmm::SwapCode, [1227](#)

- BALCPPProtect
 - gdcm::Anonymizer, [139](#)
- Base64
 - gdcm::Base64, [222](#)
- BaseQuery
 - gdcm::BaseQuery, [233](#)
- BaseRootQuery
 - gdcm::BaseRootQuery, [238](#)
- BasicAnnotationBoxSOPClass
 - gdcm::UIDs, [1303](#)
- BasicApplicationLevelConfidentialityProfile
 - gdcm::Anonymizer, [139](#)
- BasicCodedEntry
 - gdcm::SegmentHelper::BasicCodedEntry, [242](#)
- BasicCodedEntryVector
 - gdcm::Segment, [1072](#)
- BasicColorImageBoxSOPClass
 - gdcm::UIDs, [1303](#)
- BasicColorPrintManagementMetaSOPClass
 - gdcm::UIDs, [1303](#)
- BasicFilmBoxSOPClass
 - gdcm::UIDs, [1303](#)
- BasicFilmSessionSOPClass
 - gdcm::UIDs, [1303](#)
- BasicGrayscaleImageBoxSOPClass
 - gdcm::UIDs, [1303](#)
- BasicGrayscalePrintManagementMetaSOPClass
 - gdcm::UIDs, [1303](#)
- BasicOffsetTable
 - gdcm::BasicOffsetTable, [247](#)
- BasicPrintImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, [1303](#)
- BasicStructuredDisplayStorage
 - gdcm::UIDs, [1310](#)
- BasicStudyContentNotificationSOPClassRetired
 - gdcm::UIDs, [1302](#)
- BasicTextSR
 - gdcm::MediaStorage, [800](#)
- BasicTextSRStorage
 - gdcm::UIDs, [1305](#)
- BasicVoiceAudioWaveformStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- Begin
 - gdcm::CSAHeaderDict, [356](#)
 - gdcm::DataSet, [391](#)
 - gdcm::Dict, [421](#)
 - gdcm::IODs, [713](#)
 - gdcm::Scanner, [1054](#)
 - gdcm::Scanner2, [1064](#)
 - gdcm::SequenceOfFragments, [1095](#)
 - gdcm::SequenceOfItems, [1103](#), [1104](#)
 - gdcm::StrictScanner, [1174](#)
 - gdcm::StrictScanner2, [1183](#)
- BigEndian
 - gdcm::SwapCode, [1227](#)
- Bitmap
 - gdcm::Bitmap, [251](#)
 - gdcm::JPEG2000Codec, [745](#)
 - gdcm::PixelFormat, [938](#)
- BitmapToBitmapFilter
 - gdcm::BitmapToBitmapFilter, [264](#)
- BitSample
 - gdcm::JPEGCodec, [760](#)
 - gdcm::LookupTable, [784](#)
- black
 - gdcm::terminal, [110](#)
- BlendingSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, [1305](#)
- blink
 - gdcm::terminal, [110](#)
- BLUE
 - gdcm::LookupTable, [779](#)
- blue
 - gdcm::terminal, [110](#)
- BOOL_FUNCTION_PFILE_PFILE_POINTER
 - gdcm, [86](#)
- BoundingBox
 - gdcm::BoxRegion, [267](#)
- BoxRegion
 - gdcm::BoxRegion, [267](#)
- BreakConnection
 - gdcm::network::ULConnectionManager, [1385](#)
- BreakConnectionNow
 - gdcm::network::ULConnectionManager, [1385](#)
- BreastImagingRelevantPatientInformationQuery
 - gdcm::UIDs, [1307](#)
- BreastProjectionXRayImageStorageForPresentation
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1309](#)
- BreastProjectionXRayImageStorageForProcessing
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1309](#)
- BreastTomosynthesisImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- bright
 - gdcm::terminal, [110](#)
- Bug List, [7](#)
- Build
 - vtkLookupTable16, [1541](#)
- ByteBuffer
 - gdcm::ByteBuffer, [271](#)
- bytes
 - gdcm::Tag, [1255](#)
- ByteSwap
 - gdcm::ByteSwapFilter, [274](#)
- ByteSwapFilter

- gdcmm::ByteSwapFilter, [274](#)
- ByteValue
 - gdcmm::ByteValue, [278](#)
- C_CANCEL_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_ECHO_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_ECHO_RSP
 - gdcmm::network::DIMSE, [440](#)
- C_FIND_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_FIND_RSP
 - gdcmm::network::DIMSE, [440](#)
- C_GET_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_GET_RSP
 - gdcmm::network::DIMSE, [440](#)
- C_MOVE_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_MOVE_RSP
 - gdcmm::network::DIMSE, [440](#)
- C_STORE_RQ
 - gdcmm::network::DIMSE, [440](#)
- C_STORE_RSP
 - gdcmm::network::DIMSE, [440](#)
- CALIBRATED
 - gdcmm::Spacing, [1150](#)
- CanCode
 - gdcmm::AudioCodec, [221](#)
 - gdcmm::Coder, [311](#)
 - gdcmm::ImageCodec, [660](#)
 - gdcmm::JPEG2000Codec, [742](#)
 - gdcmm::JPEGCodec, [755](#)
 - gdcmm::JPEGLSCodec, [764](#)
 - gdcmm::KAKADUCodec, [772](#)
 - gdcmm::PDFCodec, [916](#)
 - gdcmm::PGXCodec, [925](#)
 - gdcmm::PNMCodec, [961](#)
 - gdcmm::PVRGCodec, [1002](#)
 - gdcmm::RAWCodec, [1022](#)
 - gdcmm::RLECodec, [1044](#)
- CanDecode
 - gdcmm::AudioCodec, [221](#)
 - gdcmm::Decoder, [406](#)
 - gdcmm::DeltaEncodingCodec, [414](#)
 - gdcmm::ImageCodec, [660](#)
 - gdcmm::JPEG2000Codec, [742](#)
 - gdcmm::JPEGCodec, [755](#)
 - gdcmm::JPEGLSCodec, [764](#)
 - gdcmm::KAKADUCodec, [772](#)
 - gdcmm::PDFCodec, [916](#)
 - gdcmm::PGXCodec, [925](#)
 - gdcmm::PNMCodec, [961](#)
- gdcmm::PVRGCodec, [1002](#)
- gdcmm::RAWCodec, [1022](#)
- gdcmm::RLECodec, [1044](#)
- CanDisplay
 - gdcmm::VR, [1434](#)
- CanEmptyTag
 - gdcmm::Anonymizer, [139](#)
- CanRead
 - gdcmm::Reader, [1027](#)
- CanReadFile
 - vtkGDCMImageReader, [1448](#)
 - vtkGDCMImageReader2, [1463](#)
- CanReadImage
 - gdcmm::StreamImageReader, [1160](#)
- CanStoreLossy
 - gdcmm::TransferSyntax, [1273](#)
- CanWriteFile
 - gdcmm::StreamImageWriter, [1165](#)
- CAPI
 - gdcmm::CryptoFactory, [336](#)
- CAPICryptoFactory
 - gdcmm::CAPICryptoFactory, [285](#)
- CAPICryptographicMessageSyntax
 - gdcmm::CAPICryptographicMessageSyntax, [287](#)
- CardiacElectrophysiologyWaveformStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- CardiacRelevantPatientInformationQuery
 - gdcmm::UIDs, [1307](#)
- CEcho
 - gdcmm::CompositeNetworkFunctions, [327](#)
- CFind
 - gdcmm::CompositeNetworkFunctions, [327](#)
- Change
 - gdcmm::FileChangeTransferSyntax, [560](#)
 - gdcmm::FileDecompressLookupTable, [563](#)
 - gdcmm::FileExplicitFilter, [569](#)
 - gdcmm::ImageChangePhotometricInterpretation, [643](#)
 - gdcmm::ImageChangePlanarConfiguration, [648](#)
 - gdcmm::ImageChangeTransferSyntax, [653](#)
- ChangeFMI
 - gdcmm::FileExplicitFilter, [569](#)
- ChangeMonochrome
 - gdcmm::ImageChangePhotometricInterpretation, [643](#)
- ChangeRGB2YBR
 - gdcmm::ImageChangePhotometricInterpretation, [643](#)
- ChangeYBR2RGB
 - gdcmm::ImageChangePhotometricInterpretation, [643](#)
- CharacterDataHandler
 - gdcmm::TableReader, [1241](#)

- gdcmm::XMLDictReader, 1566
- gdcmm::XMLPrivateDictReader, 1573
- CheckDataElement
 - gdcmm::FileStreamer, 596
- CheckEvent
 - gdcmm::AnonymizeEvent, 134
 - gdcmm::DataEvent, 387
 - gdcmm::DataSetEvent, 403
 - gdcmm::Event, 534
 - gdcmm::FileNameEvent, 587
 - gdcmm::ProgressEvent, 998
- CheckFileMetaInformationOff
 - gdcmm::Writer, 1561
- CheckFileMetaInformationOn
 - gdcmm::Writer, 1561
- CheckTemplateFileName
 - gdcmm::FileStreamer, 596
- ChestCADSRStorage
 - gdcmm::UIDs, 1306
- CipherTypes
 - gdcmm::CryptographicMessageSyntax, 338
- Clamp
 - gdcmm, 91
- Clean
 - gdcmm::Cleaner, 300
- clean
 - gdcmm, 91
- Cleaner
 - gdcmm::Cleaner, 300
- CleanupUnusedBits
 - gdcmm::ImageCodec, 660
- Clear
 - gdcmm::Anonymizer, 139
 - gdcmm::Bitmap, 252
 - gdcmm::ByteValue, 278
 - gdcmm::DataElement, 373
 - gdcmm::DataSet, 392
 - gdcmm::IOD, 708
 - gdcmm::IODs, 713
 - gdcmm::Item, 724
 - gdcmm::LookupTable, 780
 - gdcmm::Macro, 788
 - gdcmm::Macros, 790
 - gdcmm::Module, 828
 - gdcmm::Modules, 835
 - gdcmm::Preamble, 964
 - gdcmm::SequenceOfFragments, 1095
 - gdcmm::SequenceOfItems, 1104
 - gdcmm::SerieHelper, 1111
 - gdcmm::Value, 1416
 - vtkGDCMMedicalImageProperties, 1483
 - vtkRTStructSetProperties, 1546
- ClearInternalUIDs
 - gdcmm::Anonymizer, 139
- ClearPrivateTags
 - gdcmm::Scanner2, 1064
 - gdcmm::StrictScanner2, 1183
- ClearPublicTags
 - gdcmm::Scanner2, 1064
 - gdcmm::StrictScanner2, 1183
- ClearSkipTags
 - gdcmm::Scanner, 1054
 - gdcmm::Scanner2, 1064
 - gdcmm::StrictScanner, 1174
 - gdcmm::StrictScanner2, 1184
- ClearTags
 - gdcmm::Scanner, 1054
 - gdcmm::StrictScanner, 1174
- Clone
 - gdcmm::BoxRegion, 268
 - gdcmm::ImageCodec, 660
 - gdcmm::JPEG2000Codec, 742
 - gdcmm::JPEGCodec, 756
 - gdcmm::JPEGLSCodec, 764
 - gdcmm::KAKADUCodec, 772
 - gdcmm::PGXCodec, 925
 - gdcmm::PNMCodec, 962
 - gdcmm::PVRGCodec, 1002
 - gdcmm::RAWCodec, 1022
 - gdcmm::Region, 1034
 - gdcmm::RLECodec, 1044
- CM
 - gdcmm::SegmentHelper::BasicCodedEntry, 243
- cMaxEventID
 - gdcmm::network, 108
- cMaxStateID
 - gdcmm::network, 108
- CMove
 - gdcmm::CompositeNetworkFunctions, 328
- CMYK
 - gdcmm::PhotometricInterpretation, 928
- Code
 - gdcmm::Coder, 311
 - gdcmm::JPEG2000Codec, 742
 - gdcmm::JPEGCodec, 756
 - gdcmm::JPEGLSCodec, 765
 - gdcmm::JSON, 768
 - gdcmm::KAKADUCodec, 772
 - gdcmm::PVRGCodec, 1003
 - gdcmm::RAWCodec, 1023
 - gdcmm::RLECodec, 1044
- CodedEntryData
 - gdcmm::Cleaner, 300
- CodeMeaning
 - gdcmm::RealWorldValueMappingContent, 1032
- CodeString
 - gdcmm::CodeString, 314, 315
- CodeValue

- gdcmm::RealWorldValueMappingContent, [1032](#)
- ColonCADSRStorage
 - gdcmm::UIDs, [1310](#)
- Color
 - gdcmm::terminal, [110](#)
- ColorArray
 - gdcmm::SurfaceHelper, [1214](#)
- ColorPaletteQueryRetrieveInformationModelFIND
 - gdcmm::UIDs, [1311](#)
- ColorPaletteQueryRetrieveInformationModelGET
 - gdcmm::UIDs, [1311](#)
- ColorPaletteQueryRetrieveInformationModelMOVE
 - gdcmm::UIDs, [1311](#)
- ColorPaletteStorage
 - gdcmm::UIDs, [1311](#)
- ColorSoftcopyPresentationStateStorageSOPClass
 - gdcmm::UIDs, [1304](#)
- Command
 - gdcmm::Command, [318](#)
- CommandDataSet
 - gdcmm::CommandDataSet, [322](#)
- CommandTypes
 - gdcmm::network::DIMSE, [440](#)
- Common Directory Reference, [57](#)
- Compatible
 - gdcmm::VM, [1428](#)
 - gdcmm::VR, [1434](#)
- Component
 - gdcmm::PersonName, [921](#)
- CompOperators
 - gdcmm, [89](#)
- CompositeInstanceRetrieveWithoutBulkDataGET
 - gdcmm::UIDs, [1310](#)
- CompositeInstanceRootRetrieveGET
 - gdcmm::UIDs, [1310](#)
- CompositeInstanceRootRetrieveMOVE
 - gdcmm::UIDs, [1310](#)
- CompositingPlanarMPRVolumetricPresentation-
StateStorage
 - gdcmm::UIDs, [1309](#)
- Comprehensive3DSRStorage
 - gdcmm::UIDs, [1310](#)
- ComprehensiveSR
 - gdcmm::MediaStorage, [800](#)
- ComprehensiveSRStorage
 - gdcmm::UIDs, [1305](#)
- ComprehensiveSRStorageTrialRetired
 - gdcmm::UIDs, [1305](#)
- CompressionTypes
 - vtkGDCMImageWriter, [1476](#)
- Compute
 - gdcmm::EquipmentManufacturer, [531](#)
 - gdcmm::MD5, [793](#)
 - gdcmm::SHA1, [1125](#)
- ComputeBoundingBox
 - gdcmm::BoxRegion, [268](#)
 - gdcmm::Region, [1034](#)
- ComputeBufferLength
 - gdcmm::ImageRegionReader, [689](#)
- ComputeByteLength
 - gdcmm::SequenceOfFragments, [1095](#)
- ComputeCSAImageHeaderInfo
 - gdcmm::SplitMosaicFilter, [1152](#)
- ComputeCSASeriesHeaderInfo
 - gdcmm::SplitMosaicFilter, [1152](#)
- ComputeDataElement
 - gdcmm::DataSet, [392](#)
- ComputeDataSetMediaStorageSOPClass
 - gdcmm::FileMetaInformation, [576](#)
- ComputeDataSetTransferSyntax
 - gdcmm::FileMetaInformation, [576](#)
- ComputeDistAlongNormal
 - gdcmm::DirectionCosines, [442](#)
- ComputedRadiographyImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- ComputeFile
 - gdcmm::MD5, [793](#)
 - gdcmm::SHA1, [1125](#)
- ComputeFileMD5
 - gdcmm::Testing, [1259](#)
- ComputeGroupLength
 - gdcmm::DataSet, [392](#)
- ComputeInterceptSlopePixelType
 - gdcmm::Rescaler, [1037](#)
- ComputeLength
 - gdcmm::ByteValue, [278](#)
 - gdcmm::Fragment, [613](#)
 - gdcmm::SequenceOfFragments, [1096](#)
 - gdcmm::SequenceOfItems, [1104](#)
- ComputeLossyFlag
 - gdcmm::Bitmap, [252](#)
- ComputeMD5
 - gdcmm::Testing, [1259](#)
- ComputeMediaStorageFromModality
 - gdcmm::ImageHelper, [675](#)
- ComputeMOSAICDimensions
 - gdcmm::SplitMosaicFilter, [1153](#)
- ComputeMOSAICImagePositionPatient
 - gdcmm::SplitMosaicFilter, [1153](#)
- ComputeMOSAICSliceNormal
 - gdcmm::SplitMosaicFilter, [1153](#)
- ComputeMOSAICSlicePosition
 - gdcmm::SplitMosaicFilter, [1153](#)
- ComputeNumberOfSurfaces
 - gdcmm::SurfaceWriter, [1224](#)
- ComputeOffsetTable
 - gdcmm::JPEGCodec, [756](#)

- ComputePixelAspectRatioFromPixelSpacing
 - gdcm::Spacing, [1150](#)
- ComputePixelTypeFromMinMax
 - gdcm::Rescaler, [1037](#)
- ComputeSpacingFromImagePositionPatient
 - gdcm::ImageHelper, [675](#)
- ComputeTargetMediaStorage
 - gdcm::ImageWriter, [696](#)
- ComputeVR
 - gdcm::DataSetHelper, [404](#)
- ComputeZSpacing
 - gdcm::IPPSorter, [719](#)
- ConcatenatePDVBlobs
 - gdcm::network::PresentationDataValue, [980](#)
- ConcatenatePDVBlobsAsExplicit
 - gdcm::network::PresentationDataValue, [980](#)
- CONDENSED_STYLE
 - gdcm::Printer, [985](#)
- Conditional
 - gdcm::Usage, [1405](#)
- CONSOLE
 - gdcm::terminal, [110](#)
- const
 - gdcm::SOPClassUIDToIOD, [1142](#)
- const_iterator
 - gdcm::CodeString, [313](#)
 - gdcm::LO, [775](#)
 - gdcm::String< TDelimiter, TMaxLength, TPad-Char >, [1190](#)
- const_reference
 - gdcm::CodeString, [313](#)
 - gdcm::LO, [775](#)
 - gdcm::String< TDelimiter, TMaxLength, TPad-Char >, [1190](#)
- const_reverse_iterator
 - gdcm::CodeString, [313](#)
 - gdcm::LO, [775](#)
 - gdcm::String< TDelimiter, TMaxLength, TPad-Char >, [1191](#)
- ConstCharWrapper
 - gdcm::ConstCharWrapper, [330](#)
- ConstIterator
 - gdcm::CSAHeaderDict, [355](#)
 - gdcm::DataSet, [391](#)
 - gdcm::Dict, [420](#)
 - gdcm::Scanner, [1052](#)
 - gdcm::SequenceOfFragments, [1094](#)
 - gdcm::SequenceOfItems, [1102](#)
 - gdcm::StrictScanner, [1172](#)
- Construct
 - gdcm::BaseRootQuery, [239](#)
- ConstructAbortPDU
 - gdcm::network::PDUFactory, [917](#)
- ConstructCEchoRQ
 - gdcm::network::CompositeMessageFactory, [325](#)
- ConstructCFindRQ
 - gdcm::network::CompositeMessageFactory, [325](#)
- ConstructCMoveRQ
 - gdcm::network::CompositeMessageFactory, [325](#)
- ConstructCStoreRQ
 - gdcm::network::CompositeMessageFactory, [325](#)
- ConstructCStoreRSP
 - gdcm::network::CompositeMessageFactory, [325](#)
- ConstructFromString
 - gdcm::DPath, [451](#)
 - gdcm::TagPath, [1256](#)
- ConstructFromTagList
 - gdcm::TagPath, [1256](#)
- ConstructNAction
 - gdcm::network::NormalizedMessageFactory, [865](#)
- ConstructNCreate
 - gdcm::network::NormalizedMessageFactory, [865](#)
- ConstructNDelete
 - gdcm::network::NormalizedMessageFactory, [865](#)
- ConstructNEventReport
 - gdcm::network::NormalizedMessageFactory, [866](#)
- ConstructNGet
 - gdcm::network::NormalizedMessageFactory, [866](#)
- ConstructNSet
 - gdcm::network::NormalizedMessageFactory, [866](#)
- ConstructorType
 - gdcm::Dicts, [436](#)
- ConstructPDU
 - gdcm::network::PDUFactory, [917](#)
- ConstructPDV
 - gdcm::network::BaseCompositeMessage, [226](#)
 - gdcm::network::BaseNormalizedMessage, [228](#)
 - gdcm::network::CEchoRQ, [291](#)
 - gdcm::network::CFindRQ, [295](#)
 - gdcm::network::CMoveRQ, [307](#)
 - gdcm::network::CStoreRQ, [362](#)
 - gdcm::network::CStoreRSP, [364](#)
 - gdcm::network::NActionRQ, [847](#)
 - gdcm::network::NCreateRQ, [850](#)
 - gdcm::network::NDeleteRQ, [853](#)
 - gdcm::network::NEventReportRQ, [859](#)
 - gdcm::network::NGetRQ, [862](#)
 - gdcm::network::NSetRQ, [870](#)
- ConstructPDVByDataSet
 - gdcm::network::CEchoRSP, [292](#)
 - gdcm::network::CFindCancelRQ, [294](#)
 - gdcm::network::CFindRSP, [297](#)
 - gdcm::network::CMoveCancelRq, [305](#)
 - gdcm::network::CMoveRSP, [308](#)
 - gdcm::network::NActionRSP, [848](#)
 - gdcm::network::NCreateRSP, [851](#)
 - gdcm::network::NDeleteRSP, [854](#)
 - gdcm::network::NEventReportRSP, [860](#)

- gdcmm::network::NGetRSP, [863](#)
- gdcmm::network::NSetRSP, [871](#)
- ConstructQuery
 - gdcmm::CompositeNetworkFunctions, [328](#), [329](#)
 - gdcmm::NormalizedNetworkFunctions, [867](#)
- ConstructReleasePDU
 - gdcmm::network::PDUFactory, [917](#)
- ContentAssessmentResultsStorage
 - gdcmm::UIDs, [1310](#)
- Convert
 - gdcmm::DictConverter, [425](#)
 - gdcmm::ImageConverter, [669](#)
- ConvertRGBToPaletteColor
 - gdcmm::IconImageGenerator, [625](#)
- ConvertToCXX
 - gdcmm::DictConverter, [426](#)
- ConvertToUNC
 - gdcmm::System, [1231](#)
- ConvertToXML
 - gdcmm::DictConverter, [426](#)
- CornealTopographyMapStorage
 - gdcmm::UIDs, [1310](#)
- CORONAL
 - gdcmm::Orientation, [887](#)
- Create
 - gdcmm::Preamble, [964](#)
- CreateCEchoPDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateCFindPDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateCMovePDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateCMSProvider
 - gdcmm::CAPICryptoFactory, [285](#)
 - gdcmm::CryptoFactory, [337](#)
 - gdcmm::OpenSSLCryptoFactory, [877](#)
 - gdcmm::OpenSSLP7CryptoFactory, [882](#)
- CreateCStoreRQPDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateCStoreRSPDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateDefaultUniqueSeriesIdentifier
 - gdcmm::SerieHelper, [1112](#)
- CreateNActionPDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateNCreatePDU
 - gdcmm::network::PDUFactory, [918](#)
- CreateNDeletePDU
 - gdcmm::network::PDUFactory, [919](#)
- CreateNEventReportPDU
 - gdcmm::network::PDUFactory, [919](#)
- CreateNGetPDU
 - gdcmm::network::PDUFactory, [919](#)
- CreateNSetPDU
 - gdcmm::network::PDUFactory, [919](#)
- CreateUniqueSeriesIdentifier
 - gdcmm::SerieHelper, [1112](#)
- Cross
 - gdcmm::DirectionCosines, [442](#)
- CrossDot
 - gdcmm::DirectionCosines, [442](#)
- CryptoFactory
 - gdcmm::CryptoFactory, [336](#)
- CryptographicMessageSyntax
 - gdcmm::CryptographicMessageSyntax, [338](#), [339](#)
- CryptoLib
 - gdcmm::CryptoFactory, [336](#)
- CS
 - gdcmm::VR, [1432](#)
- CSAElement
 - gdcmm::CSAElement, [343](#)
- CSAHeader
 - gdcmm::CSAHeader, [351](#)
 - gdcmm::DataSet, [400](#)
- CSAHeaderDict
 - gdcmm::CSAHeaderDict, [355](#)
- CSAHeaderDictEntry
 - gdcmm::CSAHeaderDictEntry, [358](#)
- CSAHeaderType
 - gdcmm::CSAHeader, [350](#)
- CSANonImageStorage
 - gdcmm::MediaStorage, [800](#)
- CSComp
 - gdcmm, [86](#)
- CSD
 - gdcmm::SegmentHelper::BasicCodedEntry, [243](#)
- CStore
 - gdcmm::CompositeNetworkFunctions, [329](#)
- CSV
 - gdcmm::SegmentHelper::BasicCodedEntry, [243](#)
- CT_private_ELE
 - gdcmm::TransferSyntax, [1272](#)
- CTDefinedProcedureProtocolStorage
 - gdcmm::UIDs, [1310](#)
- CTImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- CTPerformedProcedureProtocolStorage
 - gdcmm::UIDs, [1310](#)
- Curve
 - gdcmm::Curve, [366](#)
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1471](#)
- Curves
 - gdcmm::Pixmap, [945](#)
- CV
 - gdcmm::SegmentHelper::BasicCodedEntry, [244](#)
- CXX

- gdcmm::Printer, 985
- cyan
 - gdcmm::terminal, 110
- DA
 - gdcmm::VR, 1432
- DAComp
 - gdcmm, 86
- DataDictionary Directory Reference, 59
- DataElement
 - gdcmm::DataElement, 373
 - gdcmm::Value, 1417
- DataElementSet
 - gdcmm::DataSet, 391
- DataElementType
 - gdcmm::ModuleEntry, 833
- DataEvent
 - gdcmm::DataEvent, 386, 387
- DataField
 - gdcmm::CSAElement, 347
- DataPtr
 - gdcmm::CSAElement, 342
- DATASET_FORMAT
 - gdcmm::CSAHeader, 350
- DataSetEvent
 - gdcmm::DataSetEvent, 402
- DataSetHandled
 - gdcmm::network::ULConnectionCallback, 1379
- DataSetHandles
 - gdcmm::network::ULConnectionCallback, 1379
- DataSetMS
 - gdcmm::FileMetaInformation, 581
- DataSetTS
 - gdcmm::FileMetaInformation, 581
- DataStructureAndEncodingDefinition Directory Reference, 60
- DataWasPassed
 - vtkImageMapToColors16, 1528
- dCor
 - gdcmm::MrProtocol::Vector3, 1418
- DebugOff
 - gdcmm::Trace, 1265
- DebugOn
 - gdcmm::Trace, 1265
- Decode
 - gdcmm::AudioCodec, 221
 - gdcmm::Base64, 223
 - gdcmm::Curve, 366
 - gdcmm::Decoder, 406
 - gdcmm::DeltaEncodingCodec, 414
 - gdcmm::ImageCodec, 660
 - gdcmm::JPEG2000Codec, 742
 - gdcmm::JPEGCodec, 756
 - gdcmm::JPEGLSCodec, 765
 - gdcmm::JSON, 768
 - gdcmm::KAKADUCodec, 773
 - gdcmm::LookupTable, 780
 - gdcmm::PDFCodec, 916
 - gdcmm::PVRGCodec, 1003
 - gdcmm::RAWCodec, 1023
 - gdcmm::RLECodec, 1045
- Decode8
 - gdcmm::LookupTable, 780
- DecodeByStreams
 - gdcmm::Decoder, 406
 - gdcmm::ImageCodec, 661
 - gdcmm::JPEG12Codec, 731
 - gdcmm::JPEG16Codec, 736
 - gdcmm::JPEG2000Codec, 743
 - gdcmm::JPEG8Codec, 750
 - gdcmm::JPEGCodec, 756
 - gdcmm::RAWCodec, 1023
 - gdcmm::RLECodec, 1045
- DecodeBytes
 - gdcmm::RAWCodec, 1023
- DecodeExtent
 - gdcmm::JPEG2000Codec, 743
 - gdcmm::JPEGCodec, 757
 - gdcmm::JPEGLSCodec, 765
 - gdcmm::RLECodec, 1045
- Decompress
 - gdcmm::Overlay, 893
- Decrypt
 - gdcmm::CAPICryptographicMessageSyntax, 287
 - gdcmm::CryptographicMessageSyntax, 339
 - gdcmm::OpenSSLCryptographicMessageSyntax, 879
 - gdcmm::OpenSSLP7CryptographicMessageSyntax, 884
- DeepCopy
 - vtkRTStructSetProperties, 1546
- DEFAULT
 - gdcmm::CryptoFactory, 336
- Default
 - gdcmm::FileMetaInformation, 576
- DefinedProcedureProtocolInformationModelFIND
 - gdcmm::UIDs, 1311
- DefinedProcedureProtocolInformationModelGET
 - gdcmm::UIDs, 1311
- DefinedProcedureProtocolInformationModelMOVE
 - gdcmm::UIDs, 1311
- DefinedTerms
 - gdcmm::DefinedTerms, 407
- DefinePixelExtent
 - gdcmm::StreamImageReader, 1160
 - gdcmm::StreamImageWriter, 1165
- DefineProperBufferLength
 - gdcmm::StreamImageReader, 1160

- gdcmm::StreamImageWriter, [1165](#)
- DeflatedExplicitVRLittleEndian
 - gdcmm::TransferSyntax, [1271](#)
 - gdcmm::UIDs, [1301](#)
- DeflatedImageFrameCompression
 - gdcmm::TransferSyntax, [1272](#)
- DeformableSpatialRegistrationStorage
 - gdcmm::UIDs, [1305](#)
- Defs
 - gdcmm::Defs, [408](#)
- DeleteDirectory
 - gdcmm::System, [1231](#)
- DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [414](#)
- Deprecated List, [5](#)
- Derive
 - gdcmm::FileDerivation, [566](#)
- DES3_CIPHER
 - gdcmm::CryptographicMessageSyntax, [338](#)
- Description
 - gdcmm::ModuleEntry, [831](#)
- DescriptionField
 - gdcmm::ModuleEntry, [833](#)
- DetachedInterpretationManagementSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedPatientManagementMetaSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedPatientManagementSOPClass
 - gdcmm::MediaStorage, [800](#)
- DetachedPatientManagementSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedResultsManagementMetaSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedResultsManagementSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedStudyManagementMetaSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedStudyManagementSOPClass
 - gdcmm::MediaStorage, [800](#)
- DetachedStudyManagementSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetachedVisitManagementSOPClass
 - gdcmm::MediaStorage, [800](#)
- DetachedVisitManagementSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- DetailSRStorageTrialRetired
 - gdcmm::UIDs, [1305](#)
- DETECTOR
 - gdcmm::Spacing, [1150](#)
- DetermineEventByPDU
 - gdcmm::network::PDUFactory, [919](#)
- dicomAETitle
 - gdcmm::UIDs, [1307](#)
- dicomApplicationCluster
 - gdcmm::UIDs, [1307](#)
- DICOMApplicationContextName
 - gdcmm::UIDs, [1303](#)
- dicomAssociationAcceptor
 - gdcmm::UIDs, [1307](#)
- dicomAssociationInitiator
 - gdcmm::UIDs, [1307](#)
- dicomAuthorizedNodeCertificateReference
 - gdcmm::UIDs, [1307](#)
- dicomConfigurationRoot
 - gdcmm::UIDs, [1308](#)
- DICOMContentMappingResource
 - gdcmm::UIDs, [1311](#)
- DICOMControlledTerminology
 - gdcmm::UIDs, [1303](#)
- dicomDescription
 - gdcmm::UIDs, [1307](#)
- dicomDevice
 - gdcmm::UIDs, [1308](#)
- dicomDeviceName
 - gdcmm::UIDs, [1307](#)
- dicomDeviceSerialNumber
 - gdcmm::UIDs, [1307](#)
- dicomDevicesRoot
 - gdcmm::UIDs, [1308](#)
- DICOMDIR
 - gdcmm::DICOMDIR, [415](#)
- DICOMDIRGenerator
 - gdcmm::DICOMDIRGenerator, [417](#)
- dicomHostname
 - gdcmm::UIDs, [1307](#)
- dicomInstalled
 - gdcmm::UIDs, [1307](#)
- dicomInstitutionAddress
 - gdcmm::UIDs, [1307](#)
- dicomInstitutionDepartmentName
 - gdcmm::UIDs, [1307](#)
- dicomInstitutionName
 - gdcmm::UIDs, [1307](#)
- dicomIssuerOfPatientID
 - gdcmm::UIDs, [1307](#)
- dicomManufacturer
 - gdcmm::UIDs, [1307](#)
- dicomManufacturerModelName
 - gdcmm::UIDs, [1307](#)
- dicomNetworkAE
 - gdcmm::UIDs, [1308](#)
- dicomNetworkConnection
 - gdcmm::UIDs, [1308](#)
- dicomNetworkConnectionReference
 - gdcmm::UIDs, [1307](#)
- dicomPort
 - gdcmm::UIDs, [1307](#)

- dicomPreferredCalledAETitle
 - gdcm::UIDs, [1307](#)
- dicomPreferredCallingAETitle
 - gdcm::UIDs, [1307](#)
- dicomPrimaryDeviceType
 - gdcm::UIDs, [1307](#)
- dicomRelatedDeviceReference
 - gdcm::UIDs, [1307](#)
- dicomSoftwareVersion
 - gdcm::UIDs, [1307](#)
- dicomSOPClass
 - gdcm::UIDs, [1307](#)
- dicomStationName
 - gdcm::UIDs, [1307](#)
- dicomSupportedCharacterSet
 - gdcm::UIDs, [1307](#)
- dicomThisNodeCertificateReference
 - gdcm::UIDs, [1307](#)
- dicomTLSCyphersuite
 - gdcm::UIDs, [1307](#)
- dicomTransferCapability
 - gdcm::UIDs, [1308](#)
- dicomTransferRole
 - gdcm::UIDs, [1307](#)
- dicomTransferSyntax
 - gdcm::UIDs, [1307](#)
- DICOMUIDRegistry
 - gdcm::UIDs, [1303](#)
- dicomUniqueAETitle
 - gdcm::UIDs, [1308](#)
- dicomUniqueAETitlesRegistryRoot
 - gdcm::UIDs, [1308](#)
- dicomVendorData
 - gdcm::UIDs, [1307](#)
- DICOS2DAITStorage
 - gdcm::UIDs, [1310](#)
- DICOS3DAITStorage
 - gdcm::UIDs, [1310](#)
- DICOSCTImageStorage
 - gdcm::UIDs, [1310](#)
- DICOSDigitalXRayImageStorageForPresentation
 - gdcm::UIDs, [1310](#)
- DICOSDigitalXRayImageStorageForProcessing
 - gdcm::UIDs, [1310](#)
- DICOSQuadrupoleResonanceQRStorage
 - gdcm::UIDs, [1310](#)
- DICOSThreatDetectionReportStorage
 - gdcm::UIDs, [1310](#)
- Dict
 - gdcm::Dict, [421](#)
 - gdcm::DictEntry, [431](#)
- DICT_DEBUG
 - gdcm::DictConverter, [425](#)
- DICT_DEFAULT
 - gdcm::DictConverter, [425](#)
- DICT_XML
 - gdcm::DictConverter, [425](#)
- DictConverter
 - gdcm::DictConverter, [425](#)
- DictEntry
 - gdcm::DictEntry, [429](#)
- DictPrinter
 - gdcm::DictPrinter, [434](#)
- Dicts
 - gdcm::CSAHeaderDict, [357](#)
 - gdcm::Dict, [423](#)
 - gdcm::Dicts, [436](#)
 - gdcm::PrivateDict, [989](#)
- difference_type
 - gdcm::CodeString, [313](#)
 - gdcm::LO, [775](#)
 - gdcm::String< TDelimiter, TMaxLength, TPad-Char >, [1191](#)
- DigitalIntraoralXRayImageStorageForPresentation
 - gdcm::UIDs, [1304](#)
- DigitalIntraoralXRayImageStorageForPresentation
 - gdcm::MediaStorage, [799](#)
- DigitalIntraoralXRayImageStorageForProcessing
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- DigitalMammographyImageStorageForPresentation
 - gdcm::MediaStorage, [799](#)
- DigitalMammographyImageStorageForProcessing
 - gdcm::MediaStorage, [799](#)
- DigitalMammographyXRayImageStorageForPresentation
 - gdcm::UIDs, [1304](#)
- DigitalMammographyXRayImageStorageForProcessing
 - gdcm::UIDs, [1304](#)
- DigitalXRayImageStorageForPresentation
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- DigitalXRayImageStorageForProcessing
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- dim
 - gdcm::terminal, [110](#)
- Dimensions
 - gdcm::Bitmap, [261](#)
 - gdcm::ImageCodec, [667](#)
- DirCosTolerance
 - gdcm::IPPSorter, [719](#)
- DirectionCosines
 - gdcm::DirectionCosines, [441](#)
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1471](#)
- Directory

- gdcmm::Directory, 445
- DisplaySystemSOPClass
 - gdcmm::UIDs, 1308
- DisplaySystemSOPInstance
 - gdcmm::UIDs, 1308
- DoByteSwap
 - gdcmm::ImageCodec, 661
- DoIconImage
 - gdcmm::PixmapWriter, 956
- DoInvertMonochrome
 - gdcmm::ImageCodec, 661
- DoOverlayCleanup
 - gdcmm::ImageCodec, 661
- DoPaddedCompositePixelCode
 - gdcmm::ImageCodec, 661
- DoPlanarConfiguration
 - gdcmm::ImageCodec, 661
- doround
 - gdcmm, 92
- DoSimpleCopy
 - gdcmm::ImageCodec, 662
- Dot
 - gdcmm::DirectionCosines, 442
- DoYBR
 - gdcmm::ImageCodec, 662
- DoYBRFull422
 - gdcmm::ImageCodec, 662
- DPath
 - gdcmm::DPath, 451
- DropDuplicatePositions
 - gdcmm::IPPSorter, 719
- DS
 - gdcmm::VR, 1432
- dSag
 - gdcmm::MrProtocol::Vector3, 1418
- DT
 - gdcmm::VR, 1432
- DTCComp
 - gdcmm, 87
- dTra
 - gdcmm::MrProtocol::Vector3, 1418
- Dumper
 - gdcmm::Dumper, 455
- DuplicateAttributeError
 - gdcmm::Parser, 902
- eAABORTPDUReturnedOpen
 - gdcmm::network, 107
- eAABORTRequest
 - gdcmm::network, 107
- eAASSOCIATE_RQPDUreceived
 - gdcmm::network, 107
- eAASSOCIATERequestLocalUser
 - gdcmm::network, 107
- eAASSOCIATEResponseAccept
 - gdcmm::network, 107
- eAASSOCIATEResponseReject
 - gdcmm::network, 107
- eArabic
 - gdcmm, 89
- eARELEASE_RPPDUReceived
 - gdcmm::network, 107
- eARELEASE_RQPDUReceivedOpen
 - gdcmm::network, 107
- eARELEASERequest
 - gdcmm::network, 107
- eARELEASEResponse
 - gdcmm::network, 107
- eARTIMTimerExpired
 - gdcmm::network, 107
- eASSOCIATE_ACPDUreceived
 - gdcmm::network, 107
- eASSOCIATE_RJPDUreceived
 - gdcmm::network, 107
- ECG12leadWaveformStorage
 - gdcmm::UIDs, 1304
- ECharSet
 - gdcmm, 89
- eCreateMMPS
 - gdcmm, 90
- eCyrillic
 - gdcmm, 89
- EddyCurrentImageStorage
 - gdcmm::UIDs, 1310
- EddyCurrentMultiframeImageStorage
 - gdcmm::UIDs, 1310
- EDGE
 - gdcmm::MeshPrimitive, 815
- eEventDoesNotExist
 - gdcmm::network, 107
- EEEventID
 - gdcmm::network, 107
- eFind
 - gdcmm, 90
- eGB18030
 - gdcmm, 89
- eGreek
 - gdcmm, 89
- eHebrew
 - gdcmm, 89
- eImage
 - gdcmm, 90
- eJapanese
 - gdcmm, 89
- eJapaneseKanjiMultibyte
 - gdcmm, 89
- eJapaneseSupplementaryKanjiMultibyte
 - gdcmm, 89

- eKoreanHangulHanjaMultibyte
 - gdcm, [89](#)
- eLatin1
 - gdcm, [89](#)
- eLatin2
 - gdcm, [89](#)
- eLatin3
 - gdcm, [89](#)
- eLatin4
 - gdcm, [89](#)
- eLatin5
 - gdcm, [89](#)
- Element
 - gdcm::Element< TVR, VM::VM1_n >, [469](#)
- eMove
 - gdcm, [90](#)
- Empty
 - gdcm::Anonymizer, [140](#)
 - gdcm::BoxRegion, [268](#)
 - gdcm::Cleaner, [300](#), [301](#)
 - gdcm::DataElement, [373](#)
 - gdcm::FileAnonymizer, [555](#)
 - gdcm::Region, [1034](#)
- EmptyMaskGenerator
 - gdcm::EmptyMaskGenerator, [521](#)
- EmptyWhenScrubFails
 - gdcm::Cleaner, [301](#)
- EncapsulatedCDASStorage
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1306](#)
- EncapsulatedDocument
 - gdcm::EncapsulatedDocument, [523](#)
- EncapsulatedPDFStorage
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1306](#)
- EncapsulatedSTLStorage
 - gdcm::UIDs, [1310](#)
- Encode
 - gdcm::Base64, [223](#)
- EncodeBuffer
 - gdcm::JPEG12Codec, [731](#)
 - gdcm::JPEG16Codec, [736](#)
 - gdcm::JPEG8Codec, [750](#)
 - gdcm::JPEGCodec, [757](#)
- EncodeBytes
 - gdcm::System, [1231](#)
- Encrypt
 - gdcm::CAPICryptographicMessageSyntax, [287](#)
 - gdcm::CryptographicMessageSyntax, [339](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [879](#)
 - gdcm::OpenSSL7CryptographicMessageSyntax, [884](#)
- End
 - gdcm::CSAHeaderDict, [356](#)
 - gdcm::DataSet, [392](#)
 - gdcm::Dict, [421](#)
 - gdcm::IODs, [713](#)
 - gdcm::Scanner, [1054](#)
 - gdcm::Scanner2, [1064](#)
 - gdcm::SequenceOfFragments, [1096](#)
 - gdcm::SequenceOfItems, [1104](#)
 - gdcm::StrictScanner, [1174](#)
 - gdcm::StrictScanner2, [1184](#)
- EndElement
 - gdcm::TableReader, [1241](#)
 - gdcm::XMLDictReader, [1566](#)
 - gdcm::XMLPrivateDictReader, [1573](#)
- EndElementHandler
 - gdcm::Parser, [902](#)
- EndFilter
 - gdcm::SimpleSubjectWatcher, [1132](#)
- EndWith
 - gdcm::Filename, [582](#)
- EnhancedCTImageStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- EnhancedMRColorImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1311](#)
- EnhancedMRImageStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- EnhancedPETImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1310](#)
- EnhancedSR
 - gdcm::MediaStorage, [800](#)
- EnhancedSRStorage
 - gdcm::UIDs, [1305](#)
- EnhancedUSVolumeStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- EnhancedXAImageStorage
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1305](#)
- EnhancedXRFImageStorage
 - gdcm::UIDs, [1305](#)
- ENQueryType
 - gdcm, [89](#)
- EnumeratedValues
 - gdcm::EnumeratedValues, [530](#)
- ePatient
 - gdcm, [90](#)
- ePatientRootType
 - gdcm, [90](#)
- ePDATArequest
 - gdcm::network, [107](#)

- ePDATATFPDU
 - gdcm::network, [107](#)
- EQueryLevel
 - gdcm, [90](#)
- EQueryType
 - gdcm, [90](#)
- ERootType
 - gdcm, [90](#)
- ErrorOff
 - gdcm::Trace, [1266](#)
- ErrorOn
 - gdcm::Trace, [1266](#)
- ErrorType
 - gdcm::Parser, [902](#)
- eSeries
 - gdcm, [90](#)
- eSetMMPS
 - gdcm, [90](#)
- eSta10ReleaseCollisionAc
 - gdcm::network, [108](#)
- eSta11ReleaseCollisionRq
 - gdcm::network, [108](#)
- eSta12ReleaseCollisionAcLocal
 - gdcm::network, [108](#)
- eSta13AwaitingClose
 - gdcm::network, [108](#)
- eSta1Idle
 - gdcm::network, [108](#)
- eSta2Open
 - gdcm::network, [108](#)
- eSta3WaitLocalAssoc
 - gdcm::network, [108](#)
- eSta4LocalAssocDone
 - gdcm::network, [108](#)
- eSta5WaitRemoteAssoc
 - gdcm::network, [108](#)
- eSta6TransferReady
 - gdcm::network, [108](#)
- eSta7WaitRelease
 - gdcm::network, [108](#)
- eSta8WaitLocalRelease
 - gdcm::network, [108](#)
- eSta9ReleaseCollisionRqLocal
 - gdcm::network, [108](#)
- EstablishConnection
 - gdcm::network::ULConnectionManager, [1385](#)
- EstablishConnectionMove
 - gdcm::network::ULConnectionManager, [1386](#)
- eStaDoesNotExist
 - gdcm::network, [107](#)
- EStateID
 - gdcm::network, [107](#)
- eStudy
 - gdcm, [90](#)
- eStudyRootType
 - gdcm, [90](#)
- eThai
 - gdcm, [89](#)
- eTransportConnConfirmLocal
 - gdcm::network, [107](#)
- eTransportConnectionClosed
 - gdcm::network, [107](#)
- eTransportConnIndicLocal
 - gdcm::network, [107](#)
- eUnrecognizedPDURceived
 - gdcm::network, [107](#)
- eUTF8
 - gdcm, [89](#)
- Event
 - gdcm::Event, [533](#)
- eWLMFind
 - gdcm, [90](#)
- Exception
 - gdcm::Exception, [537](#)
- Execute
 - gdcm::Command, [319](#)
 - gdcm::EmptyMaskGenerator, [521](#)
 - gdcm::MemberCommand< T >, [810](#)
 - gdcm::SimpleMemberCommand< T >, [1130](#)
- ExecuteData
 - vtkGDCMImageReader, [1448](#)
 - vtkGDCMThreadedImageReader, [1501](#)
- ExecuteInformation
 - vtkGDCMImageReader, [1448](#)
 - vtkGDCMThreadedImageReader, [1501](#)
- ExecuteQuery
 - gdcm::StringFilter, [1195](#)
- Explicit
 - gdcm::TransferSyntax, [1271](#)
- ExplicitVRBigEndian
 - gdcm::TransferSyntax, [1271](#)
 - gdcm::UIDs, [1301](#)
- ExplicitVRLittleEndian
 - gdcm::TransferSyntax, [1271](#)
 - gdcm::UIDs, [1301](#)
- Explore
 - gdcm::Directory, [446](#)
- ExtensibleSRStorage
 - gdcm::UIDs, [1310](#)
- Extract
 - gdcm::IconImageFilter, [623](#)
- ExtractIconImages
 - gdcm::IconImageFilter, [623](#)
- ExtractVeprolIconImages
 - gdcm::IconImageFilter, [623](#)
- F
 - gdcm::Printer, [987](#)

- gdcm::Reader, [1031](#)
- gdcm::Validate, [1413](#)
- gdcm::XMLPrinter, [1570](#)
- FACET
 - gdcm::MeshPrimitive, [815](#)
- FallColorPaletteSOPInstance
 - gdcm::UIDs, [1308](#)
- FD
 - gdcm::VR, [1432](#)
- Fiducials
 - gdcm::Fiducials, [547](#)
- File
 - gdcm::File, [550](#)
- FileAnonymizer
 - gdcm::FileAnonymizer, [555](#)
- FileChangeTransferSyntax
 - gdcm::FileChangeTransferSyntax, [559](#)
 - gdcm::ImageCodec, [666](#)
- FileDecompressLookupTable
 - gdcm::FileDecompressLookupTable, [563](#)
- FileDerivation
 - gdcm::FileDerivation, [565](#)
- FileExists
 - gdcm::System, [1231](#)
- FileExplicitFilter
 - gdcm::FileExplicitFilter, [569](#)
- FileIsDirectory
 - gdcm::System, [1231](#)
- FileIsSymlink
 - gdcm::System, [1232](#)
- FileList
 - gdcm, [87](#)
- FileMetaInformation
 - gdcm::FileMetaInformation, [575](#), [576](#)
- FileName
 - vtkGDCMPolyDataReader, [1489](#)
- Filename
 - gdcm::Filename, [582](#)
- filename
 - gdcm::FileWithName, [601](#)
- FileNameEvent
 - gdcm::FileNameEvent, [586](#)
- FilenameGenerator
 - gdcm::FilenameGenerator, [589](#)
- FileNameOrdering
 - gdcm::SerieHelper, [1112](#)
- FileNames
 - vtkGDCMImageReader, [1457](#)
- Filenames
 - gdcm::Sorter, [1147](#)
- FilenamesType
 - gdcm::DICOMDIRGenerator, [417](#)
 - gdcm::Directory, [445](#)
 - gdcm::FilenameGenerator, [589](#)
- FilenameType
 - gdcm::DICOMDIRGenerator, [417](#)
 - gdcm::Directory, [445](#)
 - gdcm::FilenameGenerator, [589](#)
- FileSet
 - gdcm::FileSet, [592](#)
- FileSize
 - gdcm::System, [1232](#)
- FileStreamer
 - gdcm::FileStreamer, [596](#)
- FilesType
 - gdcm::FileSet, [592](#)
- FileTime
 - gdcm::System, [1232](#)
- FileType
 - gdcm::FileSet, [592](#)
- FileWithName
 - gdcm::FileWithName, [601](#)
- Fill
 - gdcm::ByteValue, [278](#)
- FillFromDataSet
 - gdcm::FileMetaInformation, [576](#)
- FillMedicalImageInformation
 - vtkGDCMImageReader, [1448](#)
 - vtkGDCMImageReader2, [1463](#)
 - vtkGDCMPolyDataReader, [1487](#)
- FindContext
 - gdcm::network::ULConnection, [1374](#)
- FindCSAElementByName
 - gdcm::CSAHeader, [351](#)
- FindDataElement
 - gdcm::DataSet, [392](#), [393](#)
 - gdcm::Item, [724](#)
 - gdcm::SequenceOfItems, [1104](#)
- FindDictEntry
 - gdcm::PrivateDict, [988](#)
- FindMacroEntry
 - gdcm::Macro, [788](#)
- FindModuleEntryInMacros
 - gdcm::Module, [828](#)
- FindMrProtocolByName
 - gdcm::MrProtocol, [845](#)
- FindNextDataElement
 - gdcm::DataSet, [393](#)
- FindPatientRootQuery
 - gdcm::FindPatientRootQuery, [604](#)
- FindPDBelementByName
 - gdcm::PDBHeader, [912](#)
- FindStudyRootQuery
 - gdcm::FindStudyRootQuery, [608](#)
- FirstRender
 - vtkImageColorViewer, [1521](#)
- FL
 - gdcm::VR, [1432](#)

- FLOAT16
 - gdcm::PixelFormat, [933](#)
- FLOAT32
 - gdcm::PixelFormat, [933](#)
- FLOAT64
 - gdcm::PixelFormat, [933](#)
- ForceRescale
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1471](#)
- FormatDateTime
 - gdcm::System, [1232](#)
- Fragment
 - gdcm::Fragment, [613](#)
- FragmentVector
 - gdcm::SequenceOfFragments, [1094](#)
- FromString
 - gdcm::StringFilter, [1195](#)
- FUJI
 - gdcm::EquipmentManufacturer, [531](#)
- FujiPrivateCRImageStorage
 - gdcm::MediaStorage, [801](#)
- FujiPrivateMammoCRImageStorage
 - gdcm::MediaStorage, [801](#)
- gdcm, [71](#)
 - add1, [91](#)
 - AECComp, [86](#)
 - ASComp, [86](#)
 - backslash, [91](#)
 - BOOL_FUNCTION_PFILE_PFILE_POINTER, [86](#)
 - Clamp, [91](#)
 - clean, [91](#)
 - CompOperators, [89](#)
 - CSComp, [86](#)
 - DAComp, [86](#)
 - doround, [92](#)
 - DTComp, [87](#)
 - eArabic, [89](#)
 - ECharSet, [89](#)
 - eCreateMMPS, [90](#)
 - eCyrillic, [89](#)
 - eFind, [90](#)
 - eGB18030, [89](#)
 - eGreek, [89](#)
 - eHebrew, [89](#)
 - eImage, [90](#)
 - eJapanese, [89](#)
 - eJapaneseKanjiMultibyte, [89](#)
 - eJapaneseSupplementaryKanjiMultibyte, [89](#)
 - eKoreanHangulHanjaMultibyte, [89](#)
 - eLatin1, [89](#)
 - eLatin2, [89](#)
 - eLatin3, [89](#)
 - eLatin4, [89](#)
 - eLatin5, [89](#)
 - eMove, [90](#)
 - ENQueryType, [89](#)
 - ePatient, [90](#)
 - ePatientRootType, [90](#)
 - EQueryLevel, [90](#)
 - EQueryType, [90](#)
 - ERootType, [90](#)
 - eSeries, [90](#)
 - eSetMMPS, [90](#)
 - eStudy, [90](#)
 - eStudyRootType, [90](#)
 - eThai, [89](#)
 - eUTF8, [89](#)
 - eWLMFind, [90](#)
 - FileList, [87](#)
 - GDCM_DIFFERENT, [89](#)
 - GDCM_EQUAL, [89](#)
 - GDCM_GREATER, [89](#)
 - GDCM_GREATEROREQUAL, [89](#)
 - GDCM_LESS, [89](#)
 - GDCM_LESSEQUAL, [89](#)
 - GetVRFromTag, [92](#)
 - GlobalInstance, [102](#)
 - IconImage, [87](#)
 - LD_ALL, [91](#)
 - LD_NOSEQ, [91](#)
 - LD_NOSHADOW, [91](#)
 - LD_NOSHADOWSEQ, [91](#)
 - LOComp, [87](#)
 - LodModeType, [90](#)
 - LTCComp, [87](#)
 - MacroEntry, [87](#)
 - NestedMacroEntries, [87](#)
 - operator!=, [92](#)
 - operator<<, [92–101](#)
 - operator>>, [101](#)
 - operator==, [101](#)
 - PNComp, [87](#)
 - Round, [101](#)
 - roundat, [101](#)
 - SHComp, [88](#)
 - STComp, [88](#)
 - TMComp, [88](#)
 - UCComp, [88](#)
 - UIComp, [88](#)
 - URComp, [88](#)
 - UTComp, [88](#)
 - x16printf, [102](#)
- GDCM Documentation, [1](#)
- gdcm::AbortEvent, [128](#)
- gdcm::AnonymizeEvent, [131](#)
 - ~AnonymizeEvent, [133](#)

- AnonymizeEvent, [133](#)
- CheckEvent, [134](#)
- GetEventName, [134](#)
- GetTag, [134](#)
- MakeObject, [134](#)
- operator=, [134](#)
- Self, [133](#)
- SetTag, [134](#)
- Superclass, [133](#)
- gdcmm::Anonymizer, [135](#)
 - ~Anonymizer, [138](#)
 - Anonymizer, [138](#)
 - BALCPPProtect, [139](#)
 - BasicApplicationLevelConfidentialityProfile, [139](#)
 - CanEmptyTag, [139](#)
 - Clear, [139](#)
 - ClearInternalUIDs, [139](#)
 - Empty, [140](#)
 - GetBasicApplicationLevelConfidentialityProfileAttributes, [140](#)
 - GetCryptographicMessageSyntax, [140](#)
 - GetFile, [140](#)
 - New, [141](#)
 - RecurseDataSet, [141](#)
 - Remove, [141](#)
 - RemoveGroupLength, [141](#)
 - RemovePrivateTags, [141](#)
 - RemoveRetired, [142](#)
 - Replace, [142](#)
 - SetCryptographicMessageSyntax, [143](#)
 - SetFile, [143](#)
- gdcmm::AnyEvent, [144](#)
- gdcmm::ApplicationEntity, [147](#)
 - Internal, [149](#)
 - IsValid, [148](#)
 - MaxLength, [149](#)
 - MaxNumberOfComponents, [149](#)
 - Padding, [149](#)
 - Print, [148](#)
 - Separator, [149](#)
 - SetBlob, [148](#)
 - Squeeze, [148](#)
- gdcmm::ASN1, [155](#)
 - ~ASN1, [156](#)
 - ASN1, [156](#)
 - operator=, [156](#)
 - ParseDump, [156](#)
 - ParseDumpFile, [156](#)
 - TestPBKDF2, [157](#)
- gdcmm::Attribute< Group, Element, TVR, TVM >, [158](#)
 - ArrayType, [161](#)
 - GDCM_STATIC_ASSERT, [161](#)
 - GetAsDataElement, [162](#)
 - GetDictVM, [162](#)
 - GetDictVR, [162](#)
 - GetNumberOfValues, [162](#)
 - GetTag, [162](#)
 - GetValue, [163](#)
 - GetValues, [163](#)
 - GetVM, [163](#)
 - GetVR, [164](#)
 - Internal, [167](#)
 - operator!=, [164](#)
 - operator<, [164](#)
 - operator==, [164](#)
 - operator[], [164](#), [165](#)
 - Print, [165](#)
 - Set, [165](#)
 - SetByteValue, [165](#)
 - SetByteValueNoSwap, [165](#)
 - SetFromDataElement, [166](#)
 - SetFromDataSet, [166](#)
 - SetValue, [166](#)
 - SetValues, [167](#)
 - VMType, [161](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [168](#)
 - ArrayType, [170](#)
 - GDCM_STATIC_ASSERT, [171](#)
 - GetAsDataElement, [171](#)
 - GetDictVM, [172](#)
 - GetDictVR, [172](#)
 - GetNumberOfValues, [172](#)
 - GetTag, [172](#)
 - GetValue, [172](#)
 - GetValues, [172](#)
 - GetVM, [173](#)
 - GetVR, [173](#)
 - Internal, [175](#)
 - operator!=, [173](#)
 - operator<, [173](#)
 - operator==, [173](#)
 - operator[], [173](#)
 - Print, [174](#)
 - Set, [174](#)
 - SetByteValue, [174](#)
 - SetByteValueNoSwap, [174](#)
 - SetFromDataElement, [174](#)
 - SetFromDataSet, [175](#)
 - SetValue, [175](#)
 - SetValues, [175](#)
 - VMType, [171](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [176](#)
 - ArrayType, [178](#)
 - GDCM_STATIC_ASSERT, [178](#)
 - GetAsDataElement, [178](#)

- GetDictVM, [178](#)
- GetDictVR, [178](#)
- GetNumberOfValues, [178](#)
- GetTag, [179](#)
- GetValue, [179](#)
- GetValues, [179](#)
- GetVM, [179](#)
- GetVR, [179](#)
- Internal, [181](#)
- operator!=, [179](#)
- operator<, [179](#)
- operator==, [179](#)
- operator[], [179](#)
- Print, [180](#)
- Set, [180](#)
- SetByteValue, [180](#)
- SetByteValueNoSwap, [180](#)
- SetFromDataElement, [180](#)
- SetFromDataSet, [180](#)
- SetValue, [180](#)
- SetValues, [180](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n
>, [181](#)
- ArrayType, [183](#)
- GDCM_STATIC_ASSERT, [184](#)
- GetAsDataElement, [184](#)
- GetDictVM, [184](#)
- GetDictVR, [184](#)
- GetNumberOfValues, [184](#)
- GetTag, [184](#)
- GetValue, [184](#)
- GetValues, [184](#)
- GetVM, [184](#)
- GetVR, [184](#)
- Internal, [186](#)
- operator!=, [185](#)
- operator<, [185](#)
- operator==, [185](#)
- operator[], [185](#)
- Print, [185](#)
- Set, [185](#)
- SetByteValue, [185](#)
- SetByteValueNoSwap, [185](#)
- SetFromDataElement, [186](#)
- SetFromDataSet, [186](#)
- SetValue, [186](#)
- SetValues, [186](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n gdcmm::Attribute< Group, Element, TVR, VM::VM2_n
>, [187](#)
- ~Attribute, [189](#)
- ArrayType, [189](#)
- Attribute, [189](#)
- GDCM_STATIC_ASSERT, [189](#)
- GetAsDataElement, [190](#)
- GetDictVM, [190](#)
- GetDictVR, [190](#)
- GetNumberOfValues, [190](#)
- GetTag, [190](#)
- GetValue, [190](#), [191](#)
- GetValues, [191](#)
- GetVM, [191](#)
- GetVR, [191](#)
- operator!=, [191](#)
- operator<, [191](#)
- operator==, [192](#)
- operator[], [192](#)
- Print, [192](#)
- Set, [192](#)
- SetByteValue, [192](#)
- SetByteValueNoSwap, [193](#)
- SetFromDataElement, [193](#)
- SetFromDataSet, [193](#)
- SetNumberOfValues, [193](#)
- SetValue, [193](#), [194](#)
- SetValues, [194](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n
>, [194](#)
- ArrayType, [198](#)
- GDCM_STATIC_ASSERT, [198](#)
- GetAsDataElement, [198](#)
- GetDictVM, [198](#)
- GetDictVR, [198](#)
- GetNumberOfValues, [198](#)
- GetTag, [198](#)
- GetValue, [199](#)
- GetValues, [199](#)
- GetVM, [199](#)
- GetVR, [199](#)
- Internal, [201](#)
- operator!=, [199](#)
- operator<, [199](#)
- operator==, [199](#)
- operator[], [199](#)
- Print, [200](#)
- Set, [200](#)
- SetByteValue, [200](#)
- SetByteValueNoSwap, [200](#)
- SetFromDataElement, [200](#)
- SetFromDataSet, [200](#)
- SetValue, [200](#)
- SetValues, [200](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n gdcmm::Attribute< Group, Element, TVR, VM::VM2_n
>, [201](#)
- ArrayType, [203](#)
- GDCM_STATIC_ASSERT, [204](#)
- GetAsDataElement, [204](#)
- GetDictVM, [204](#)
- GetDictVR, [204](#)

- GetNumberOfValues, [204](#)
- GetTag, [204](#)
- GetValue, [204](#)
- GetValues, [204](#)
- GetVM, [204](#)
- GetVR, [204](#)
- Internal, [206](#)
- operator!=, [205](#)
- operator<, [205](#)
- operator==, [205](#)
- operator[], [205](#)
- Print, [205](#)
- Set, [205](#)
- SetByteValue, [205](#)
- SetByteValueNoSwap, [205](#)
- SetFromDataElement, [206](#)
- SetFromDataSet, [206](#)
- SetValue, [206](#)
- SetValues, [206](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n
 >, [207](#)
- ArrayType, [210](#)
- GDCM_STATIC_ASSERT, [210](#)
- GetAsDataElement, [210](#)
- GetDictVM, [211](#)
- GetDictVR, [211](#)
- GetNumberOfValues, [211](#)
- GetTag, [211](#)
- GetValue, [211](#)
- GetValues, [211](#)
- GetVM, [211](#)
- GetVR, [211](#)
- Internal, [213](#)
- operator!=, [211](#)
- operator<, [212](#)
- operator==, [212](#)
- operator[], [212](#)
- Print, [212](#)
- Set, [212](#)
- SetByteValue, [212](#)
- SetByteValueNoSwap, [212](#)
- SetFromDataElement, [212](#)
- SetFromDataSet, [213](#)
- SetValue, [213](#)
- SetValues, [213](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n
 >, [214](#)
- ArrayType, [216](#)
- GDCM_STATIC_ASSERT, [216](#)
- GetAsDataElement, [216](#)
- GetDictVM, [216](#)
- GetDictVR, [216](#)
- GetNumberOfValues, [216](#)
- GetTag, [217](#)
- GetValue, [217](#)
- GetValues, [217](#)
- GetVM, [217](#)
- GetVR, [217](#)
- Internal, [219](#)
- operator!=, [217](#)
- operator<, [217](#)
- operator==, [217](#)
- operator[], [217](#)
- Print, [218](#)
- Set, [218](#)
- SetByteValue, [218](#)
- SetByteValueNoSwap, [218](#)
- SetFromDataElement, [218](#)
- SetFromDataSet, [218](#)
- SetValue, [218](#)
- SetValues, [218](#)
- gdcmm::AudioCodec, [219](#)
- ~AudioCodec, [221](#)
- AudioCodec, [221](#)
- CanCode, [221](#)
- CanDecode, [221](#)
- Decode, [221](#)
- gdcmm::Base64, [222](#)
- Base64, [222](#)
- Decode, [223](#)
- Encode, [223](#)
- GetDecodeLength, [223](#)
- GetEncodeLength, [224](#)
- operator=, [224](#)
- gdcmm::BaseQuery, [231](#)
- ~BaseQuery, [233](#)
- AddQueryDataSet, [233](#)
- BaseQuery, [233](#)
- GetAbstractSyntaxUID, [233](#)
- GetQueryDataSet, [234](#)
- GetSOPInstanceUID, [234](#)
- mDataSet, [236](#)
- mSopInstanceUID, [236](#)
- Print, [234](#)
- QueryFactory, [235](#)
- SetSearchParameter, [234](#)
- SetSOPInstanceUID, [235](#)
- ValidateQuery, [235](#)
- ValidDataSet, [235](#)
- WriteHelpFile, [235](#)
- WriteQuery, [235](#)
- gdcmm::BaseRootQuery, [236](#)
- ~BaseRootQuery, [238](#)
- BaseRootQuery, [238](#)
- Construct, [239](#)
- GetQueryLevelFromQueryRoot, [239](#)
- GetQueryLevelFromString, [239](#)
- GetQueryLevelString, [239](#)

- GetTagListByLevel, 239
- InitializeDataSet, 239
- mHelpDescription, 240
- mImage, 240
- mPatient, 240
- mRootType, 240
- mSeries, 241
- mStudy, 241
- QueryFactory, 240
- ValidateQuery, 239
- gdcm::BasicOffsetTable, 244
 - BasicOffsetTable, 247
 - operator<<, 248
 - Read, 247
- gdcm::Bitmap, 248
 - ~Bitmap, 251
 - AreOverlaysInPixelData, 252
 - Bitmap, 251
 - Clear, 252
 - ComputeLossyFlag, 252
 - Dimensions, 261
 - GetBuffer, 252
 - GetBuffer2, 252
 - GetBufferLength, 252
 - GetColumns, 253
 - GetDataElement, 253
 - GetDimension, 253
 - GetDimensions, 253
 - GetLUT, 254
 - GetNeedByteSwap, 254
 - GetNumberOfDimensions, 254
 - GetPhotometricInterpretation, 254
 - GetPixelFormat, 255
 - GetPlanarConfiguration, 255
 - GetRows, 255
 - GetTransferSyntax, 255
 - ImageChangeTransferSyntax, 261
 - IsEmpty, 256
 - IsLossy, 256
 - IsTransferSyntaxCompatible, 256
 - LossyFlag, 261
 - LUT, 261
 - LUTPtr, 251
 - NeedByteSwap, 261
 - NumberOfDimensions, 261
 - PF, 262
 - PI, 262
 - PixelData, 262
 - PixmapReader, 261
 - PlanarConfiguration, 262
 - Print, 256
 - SetColumns, 256
 - SetDataElement, 256
 - SetDimension, 257
 - SetDimensions, 257
 - SetLossyFlag, 257
 - SetLUT, 257
 - SetNeedByteSwap, 258
 - SetNumberOfDimensions, 258
 - SetPhotometricInterpretation, 258
 - SetPixelFormat, 258
 - SetPlanarConfiguration, 258
 - SetRows, 259
 - SetTransferSyntax, 259
 - TryJPEG2000Codec, 259
 - TryJPEG2000Codec2, 259
 - TryJPEGCodec, 259
 - TryJPEGCodec2, 259
 - TryJPEGLSCodec, 260
 - TryKAKADUCodec, 260
 - TryPVRGCodec, 260
 - TryRAWCodec, 260
 - TryRLECodec, 260
 - TS, 262
 - UnusedBitsPresentInPixelData, 260
- gdcm::BitmapToBitmapFilter, 263
 - ~BitmapToBitmapFilter, 264
 - BitmapToBitmapFilter, 264
 - GetOutput, 264
 - GetOutputAsBitmap, 264
 - Input, 265
 - Output, 265
 - SetInput, 264
- gdcm::BoxRegion, 265
 - ~BoxRegion, 267
 - Area, 267
 - BoundingBox, 267
 - BoxRegion, 267
 - Clone, 268
 - ComputeBoundingBox, 268
 - Empty, 268
 - GetXMax, 268
 - GetXMin, 268
 - GetYMax, 268
 - GetYMin, 269
 - GetZMax, 269
 - GetZMin, 269
 - IsValid, 269
 - operator=, 269
 - Print, 269
 - SetDomain, 269
- gdcm::ByteBuffer, 270
 - ByteBuffer, 271
 - Get, 271
 - GetStart, 271
 - ShiftEnd, 271
 - UpdatePosition, 271
- gdcm::ByteSwap< T >, 271

- Swap, 272
- SwapFromSwapCodeIntoSystem, 272
- SwapRange, 272
- SwapRangeFromSwapCodeIntoSystem, 272
- SystemIsBigEndian, 273
- SystemIsLittleEndian, 273
- gdcm::ByteSwapFilter, 273
 - ~ByteSwapFilter, 274
 - ByteSwap, 274
 - ByteSwapFilter, 274
 - operator=, 274
 - SetByteSwapTag, 275
- gdcm::ByteValue, 275
 - ~ByteValue, 278
 - Append, 278
 - ByteValue, 278
 - Clear, 278
 - ComputeLength, 278
 - Fill, 278
 - GetBuffer, 279
 - GetLength, 279
 - GetPointer, 279
 - GetVoidPointer, 280
 - IsEmpty, 280
 - IsPrintable, 280
 - operator const std::vector< char > &, 280
 - operator=, 281
 - operator==, 281
 - Print, 281
 - PrintASCII, 281
 - PrintASCIIXML, 281
 - PrintGroupLength, 282
 - PrintHex, 282
 - PrintHexXML, 282
 - PrintPNXML, 282
 - Read, 282
 - SetLength, 282
 - SetLengthOnly, 283
 - Write, 283
 - WriteBuffer, 283
- gdcm::CAPICryptoFactory, 284
 - CAPICryptoFactory, 285
 - CreateCMSProvider, 285
- gdcm::CAPICryptographicMessageSyntax, 286
 - ~CAPICryptographicMessageSyntax, 287
 - CAPICryptographicMessageSyntax, 287
 - Decrypt, 287
 - Encrypt, 287
 - GetCipherType, 288
 - GetInitialized, 288
 - ParseCertificateFile, 288
 - ParseKeyFile, 288
 - SetCipherType, 288
 - SetPassword, 288
- gdcm::Cleaner, 297
 - ~Cleaner, 300
 - Clean, 300
 - Cleaner, 300
 - CodedEntryData, 300
 - Empty, 300, 301
 - EmptyWhenScrubFails, 301
 - GetFile, 301
 - New, 301
 - Preserve, 302
 - Remove, 302
 - RemoveAllGroupLength, 302
 - RemoveAllIllegal, 303
 - RemoveAllMissingPrivateCreator, 303
 - RemoveMissingPrivateCreator, 303
 - ReplaceCodeMeaning, 303
 - Scrub, 303, 304
 - SetFile, 304
- gdcm::Codec, 309
- gdcm::Coder, 310
 - ~Coder, 311
 - CanCode, 311
 - Code, 311
 - InternalCode, 311
- gdcm::CodeString, 312
 - CodeString, 314, 315
 - const_iterator, 313
 - const_reference, 313
 - const_reverse_iterator, 313
 - difference_type, 313
 - GetAsString, 315
 - IsValid, 315
 - iterator, 314
 - operator!=, 316
 - operator<<, 316
 - operator==, 316
 - pointer, 314
 - reference, 314
 - reverse_iterator, 314
 - Size, 315
 - size_type, 314
 - TrimInternal, 315
 - value_type, 314
- gdcm::Command, 316
 - ~Command, 318
 - Command, 318
 - Execute, 319
 - operator=, 319
- gdcm::CommandDataSet, 320
 - ~CommandDataSet, 322
 - CommandDataSet, 322
 - Insert, 323
 - operator<<, 324
 - Read, 323

- Replace, [323](#)
- Write, [323](#)
- gdcmm::CompositeNetworkFunctions, [325](#)
 - CEcho, [327](#)
 - CFind, [327](#)
 - CMove, [328](#)
 - ConstructQuery, [328](#), [329](#)
 - CStore, [329](#)
 - KeyValuePairArrayType, [326](#)
 - KeyValuePairType, [326](#)
- gdcmm::ConstCharWrapper, [330](#)
 - ConstCharWrapper, [330](#)
 - operator const char *, [331](#)
- gdcmm::CP246ExplicitDataElement, [331](#)
 - GetLength, [334](#)
 - Read, [334](#)
 - ReadPreValue, [334](#)
 - ReadValue, [334](#)
 - ReadWithLength, [334](#)
- gdcmm::CryptoFactory, [335](#)
 - ~CryptoFactory, [337](#)
 - CAPI, [336](#)
 - CreateCMSProvider, [337](#)
 - CryptoFactory, [336](#)
 - CryptoLib, [336](#)
 - DEFAULT, [336](#)
 - GetFactoryInstance, [337](#)
 - OPENSSL, [336](#)
 - OPENSSLP7, [336](#)
- gdcmm::CryptographicMessageSyntax, [337](#)
 - ~CryptographicMessageSyntax, [338](#)
 - AES128_CIPHER, [338](#)
 - AES192_CIPHER, [338](#)
 - AES256_CIPHER, [338](#)
 - CipherTypes, [338](#)
 - CryptographicMessageSyntax, [338](#), [339](#)
 - Decrypt, [339](#)
 - DES3_CIPHER, [338](#)
 - Encrypt, [339](#)
 - GetCipherType, [339](#)
 - operator=, [339](#)
 - ParseCertificateFile, [340](#)
 - ParseKeyFile, [340](#)
 - SetCipherType, [340](#)
 - SetPassword, [340](#)
- gdcmm::CSAElement, [341](#)
 - CSAElement, [343](#)
 - DataField, [347](#)
 - DataPtr, [342](#)
 - GetByteValue, [343](#)
 - GetKey, [343](#)
 - GetName, [343](#)
 - GetNoOfItems, [344](#)
 - GetSyngoDT, [344](#)
 - GetValue, [344](#)
 - GetVM, [344](#)
 - GetVR, [345](#)
 - IsEmpty, [345](#)
 - KeyField, [347](#)
 - NameField, [348](#)
 - NoOfItemsField, [348](#)
 - operator<, [345](#)
 - operator<<, [347](#)
 - operator=, [345](#)
 - operator==, [345](#)
 - SetByteValue, [346](#)
 - SetKey, [346](#)
 - SetName, [346](#)
 - SetNoOfItems, [346](#)
 - SetSyngoDT, [346](#)
 - SetValue, [346](#)
 - SetVM, [347](#)
 - SetVR, [347](#)
 - SyngoDTField, [348](#)
 - ValueMultiplicityField, [348](#)
 - VRField, [348](#)
- gdcmm::CSAHeader, [348](#)
 - ~CSAHeader, [351](#)
 - CSAHeader, [351](#)
 - CSAHeaderType, [350](#)
 - DATASET_FORMAT, [350](#)
 - FindCSAElementByName, [351](#)
 - GetCSADataInfo, [351](#)
 - GetCSAEEnd, [351](#)
 - GetCSAElementByName, [352](#)
 - GetCSAImageHeaderInfoTag, [352](#)
 - GetCSASeriesHeaderInfoTag, [352](#)
 - GetDataSet, [352](#)
 - GetFormat, [352](#)
 - GetInterfile, [353](#)
 - GetMrProtocol, [353](#)
 - INTERFILE, [351](#)
 - LoadFromDataElement, [353](#)
 - NOMAGIC, [350](#)
 - operator<<, [354](#)
 - Print, [353](#)
 - SV10, [350](#)
 - UNKNOWN, [350](#)
 - ZEROED_OUT, [351](#)
- gdcmm::CSAHeaderDict, [354](#)
 - AddCSAHeaderDictEntry, [356](#)
 - Begin, [356](#)
 - ConstIterator, [355](#)
 - CSAHeaderDict, [355](#)
 - Dicts, [357](#)
 - End, [356](#)
 - GetCSAHeaderDictEntry, [356](#)
 - IsEmpty, [356](#)

- Iterator, [355](#)
- LoadDefault, [356](#)
- MapCSAHeaderDictEntry, [355](#)
- operator<<, [357](#)
- operator=, [356](#)
- gdcm::CSAHeaderDictEntry, [357](#)
 - CSAHeaderDictEntry, [358](#)
 - GetDescription, [358](#)
 - GetName, [358](#)
 - GetVM, [359](#)
 - GetVR, [359](#)
 - operator<, [359](#)
 - operator<<, [360](#)
 - SetDescription, [359](#)
 - SetName, [359](#)
 - SetVM, [359](#)
 - SetVR, [359](#)
- gdcm::CSAHeaderDictException, [360](#)
- gdcm::Curve, [364](#)
 - ~Curve, [366](#)
 - Curve, [366](#)
 - Decode, [366](#)
 - GetAsPoints, [366](#)
 - GetCurveDataDescriptor, [367](#)
 - GetDataValueRepresentation, [367](#)
 - GetDimensions, [367](#)
 - GetGroup, [367](#)
 - GetNumberOfCurves, [367](#)
 - GetNumberOfPoints, [367](#)
 - GetTypeOfData, [367](#)
 - GetTypeOfDataDescription, [367](#)
 - IsEmpty, [367](#)
 - Print, [368](#)
 - SetCoordinateStartValue, [368](#)
 - SetCoordinateStepValue, [368](#)
 - SetCurve, [368](#)
 - SetCurveDataDescriptor, [368](#)
 - SetCurveDescription, [368](#)
 - SetDataValueRepresentation, [368](#)
 - SetDimensions, [368](#)
 - SetGroup, [369](#)
 - SetNumberOfPoints, [369](#)
 - SetTypeOfData, [369](#)
 - Update, [369](#)
- gdcm::DataElement, [369](#)
 - Clear, [373](#)
 - DataElement, [373](#)
 - Empty, [373](#)
 - GetByteValue, [373](#)
 - GetLength, [374](#)
 - GetSequenceOffFragments, [374](#)
 - GetTag, [374](#), [375](#)
 - GetValue, [375](#)
 - GetValueAsSQ, [375](#)
- GetVL, [376](#)
- GetVR, [376](#)
- IsEmpty, [377](#)
- IsUndefinedLength, [377](#)
- operator<, [377](#)
- operator<<, [382](#)
- operator=, [378](#)
- operator==, [378](#)
- Read, [378](#)
- ReadOrSkip, [378](#)
- ReadPreValue, [378](#)
- ReadValue, [379](#)
- ReadValueWithLength, [379](#)
- ReadWithLength, [379](#)
- SetByteValue, [379](#)
- SetTag, [380](#)
- SetValue, [380](#)
- SetValueFieldLength, [381](#)
- SetVL, [381](#)
- SetVLToUndefined, [381](#)
- SetVR, [381](#)
- TagField, [383](#)
- ValueField, [383](#)
- ValueLengthField, [383](#)
- ValuePtr, [373](#)
- VRField, [383](#)
- Write, [382](#)
- gdcm::DataElementException, [384](#)
- gdcm::DataEvent, [384](#)
 - ~DataEvent, [386](#)
 - CheckEvent, [387](#)
 - DataEvent, [386](#), [387](#)
 - GetData, [387](#)
 - GetDataLength, [387](#)
 - GetEventName, [387](#)
 - MakeObject, [387](#)
 - operator=, [387](#)
 - Self, [386](#)
 - SetData, [388](#)
 - Superclass, [386](#)
- gdcm::DataSet, [388](#)
 - Begin, [391](#)
 - Clear, [392](#)
 - ComputeDataElement, [392](#)
 - ComputeGroupLength, [392](#)
 - ConstIterator, [391](#)
 - CSAHeader, [400](#)
 - DataElementSet, [391](#)
 - End, [392](#)
 - FindDataElement, [392](#), [393](#)
 - FindNextDataElement, [393](#)
 - GetDataElement, [393](#)
 - GetDEEnd, [394](#)
 - GetDES, [394](#)

- GetLength, 394
- GetMediaStorage, 395
- GetPrivateCreator, 395
- GetPrivateTag, 395
- Insert, 395
- InsertDataElement, 395
- IsEmpty, 396
- Iterator, 391
- operator<<, 400
- operator(), 396
- operator=, 396
- operator[], 396
- Print, 396
- Read, 396
- ReadNested, 397
- ReadSelectedPrivateTags, 397
- ReadSelectedPrivateTagsWithLength, 397
- ReadSelectedTags, 397
- ReadSelectedTagsWithLength, 397
- ReadUpToTag, 398
- ReadUpToTagWithLength, 398
- ReadWithLength, 398
- Remove, 398
- Replace, 398
- ReplaceEmpty, 399
- Size, 399
- SizeType, 391
- Write, 399
- gdcm::DataSetEvent, 400
 - ~DataSetEvent, 402
 - CheckEvent, 403
 - DataSetEvent, 402
 - GetDataSet, 403
 - GetEventName, 403
 - m_DataSet, 403
 - MakeObject, 403
 - operator=, 403
 - Self, 402
 - Superclass, 402
- gdcm::DataSetHelper, 404
 - ComputeVR, 404
- gdcm::Decoder, 405
 - ~Decoder, 405
 - CanDecode, 406
 - Decode, 406
 - DecodeByStreams, 406
- gdcm::DefinedTerms, 406
 - DefinedTerms, 407
- gdcm::Defs, 407
 - ~Defs, 408
 - Defs, 408
 - GetIODFromFile, 409
 - GetIODNameFromMediaStorage, 409
 - GetIODs, 409
- GetMacros, 409
- GetModules, 409, 410
- GetTypeFromTag, 410
- Global, 411
- IsEmpty, 410
- LoadDefaults, 410
- LoadFromFile, 410
- operator=, 410
- Verify, 410, 411
- gdcm::DeltaEncodingCodec, 411
 - ~DeltaEncodingCodec, 414
 - CanDecode, 414
 - Decode, 414
 - DeltaEncodingCodec, 414
- gdcm::DICOMDIR, 415
 - DICOMDIR, 415
- gdcm::DICOMDIRGenerator, 416
 - ~DICOMDIRGenerator, 417
 - AddImageDirectoryRecord, 417
 - AddPatientDirectoryRecord, 417
 - AddSeriesDirectoryRecord, 418
 - AddStudyDirectoryRecord, 418
 - DICOMDIRGenerator, 417
 - FilenamesType, 417
 - FilenameType, 417
 - Generate, 418
 - GetFile, 418
 - GetScanner, 418
 - SetDescriptor, 418
 - SetFile, 418
 - SetFilenames, 419
 - SetRootDirectory, 419
- gdcm::Dict, 419
 - AddDictEntry, 421
 - Begin, 421
 - ConstIterator, 420
 - Dict, 421
 - Dicts, 423
 - End, 421
 - GetDictEntry, 422
 - GetDictEntryByKeyword, 422
 - GetDictEntryByName, 422
 - GetKeywordFromTag, 422
 - IsEmpty, 423
 - Iterator, 420
 - LoadDefault, 423
 - MapDictEntry, 421
 - operator<<, 423
 - operator=, 423
- gdcm::DictConverter, 424
 - ~DictConverter, 425
 - AddGroupLength, 425
 - Convert, 425
 - ConvertToCXX, 426

- ConvertToXML, [426](#)
- DICT_DEBUG, [425](#)
- DICT_DEFAULT, [425](#)
- DICT_XML, [425](#)
- DictConverter, [425](#)
- GetDictName, [426](#)
- GetInputFilename, [426](#)
- GetOutputFilename, [426](#)
- GetOutputType, [426](#)
- OutputTypes, [425](#)
- Readuint16, [426](#)
- ReadVM, [426](#)
- ReadVR, [427](#)
- SetDictName, [427](#)
- SetInputFileName, [427](#)
- SetOutputFileName, [427](#)
- SetOutputType, [427](#)
- WriteFooter, [427](#)
- WriteHeader, [427](#)
- gdcmm::DictEntry, [428](#)
 - Dict, [431](#)
 - DictEntry, [429](#)
 - GetKeyword, [429](#)
 - GetName, [429](#)
 - GetRetired, [429](#)
 - GetVM, [429](#)
 - GetVR, [429](#)
 - IsUnique, [430](#)
 - operator<<, [431](#)
 - SetElementXX, [430](#)
 - SetGroupXX, [430](#)
 - SetKeyword, [430](#)
 - SetName, [430](#)
 - SetRetired, [430](#)
 - SetVM, [431](#)
 - SetVR, [431](#)
- gdcmm::DictPrinter, [432](#)
 - ~DictPrinter, [434](#)
 - DictPrinter, [434](#)
 - Print, [434](#)
 - PrintDataElement2, [434](#)
 - PrintDataSet2, [434](#)
- gdcmm::Dicts, [435](#)
 - ~Dicts, [436](#)
 - ConstructorType, [436](#)
 - Dicts, [436](#)
 - GEMS, [436](#)
 - GetConstructorString, [437](#)
 - GetCSAHeaderDict, [437](#)
 - GetDictEntry, [437](#)
 - GetPrivateDict, [437](#)
 - GetPublicDict, [438](#)
 - Global, [438](#)
 - IsEmpty, [438](#)
 - LoadDefaults, [438](#)
 - operator<<, [438](#)
 - operator=, [438](#)
 - PHILIPS, [436](#)
 - SIEMENS, [436](#)
- gdcmm::DirectionCosines, [440](#)
 - ~DirectionCosines, [442](#)
 - ComputeDistAlongNormal, [442](#)
 - Cross, [442](#)
 - CrossDot, [442](#)
 - DirectionCosines, [441](#)
 - Dot, [442](#)
 - IsValid, [443](#)
 - Norm, [443](#)
 - Normalize, [443](#)
 - operator const double *, [443](#)
 - Print, [443](#)
 - SetFromString, [443](#)
- gdcmm::Directory, [444](#)
 - ~Directory, [445](#)
 - Directory, [445](#)
 - Explore, [446](#)
 - FilenamesType, [445](#)
 - FilenameType, [445](#)
 - GetDirectories, [446](#)
 - GetFilenames, [446](#)
 - GetToplevel, [446](#)
 - Load, [446](#)
 - operator<<, [447](#)
 - Print, [447](#)
- gdcmm::DirectoryHelper, [448](#)
 - GetCTImageSeriesUIDs, [448](#)
 - GetFilenamesFromSeriesUIDs, [448](#)
 - GetFrameOfReference, [449](#)
 - GetMRImageSeriesUIDs, [449](#)
 - GetRTStructSeriesUIDs, [449](#)
 - GetSeriesUIDsBySOPClassUID, [449](#)
 - GetSOPClassUID, [449](#)
 - GetStringValueFromTag, [449](#)
 - LoadImageFromFiles, [450](#)
 - RetrieveSOPInstanceUIDFromIndex, [450](#)
 - RetrieveSOPInstanceUIDFromZPosition, [450](#)
- gdcmm::DPath, [450](#)
 - ~DPath, [451](#)
 - ConstructFromString, [451](#)
 - DPath, [451](#)
 - IsValid, [451](#)
 - Match, [451](#)
 - operator<, [452](#)
 - operator<<, [452](#)
 - Print, [452](#)
- gdcmm::DummyValueGenerator, [452](#)
 - Generate, [453](#)
- gdcmm::Dumper, [453](#)

- ~Dumper, [455](#)
- Dumper, [455](#)
- gdcmm::Element< TVR, TVM >, [456](#)
 - GetAsDataElement, [458](#)
 - GetLength, [458](#)
 - GetValue, [459](#)
 - GetValues, [459](#)
 - GetVM, [459](#)
 - GetVR, [459](#)
 - Internal, [462](#)
 - operator[], [460](#)
 - Print, [460](#)
 - Read, [460](#)
 - Set, [460](#)
 - SetFromDataElement, [460](#)
 - SetNoSwap, [461](#)
 - SetValue, [461](#)
 - Type, [458](#)
 - Write, [461](#)
- gdcmm::Element< TVR, VM::VM1_2 >, [462](#)
 - GetAsDataElement, [465](#)
 - GetLength, [465](#)
 - GetValue, [465](#)
 - GetValues, [465](#)
 - GetVM, [465](#)
 - GetVR, [466](#)
 - Internal, [467](#)
 - operator[], [466](#)
 - Parent, [465](#)
 - Print, [466](#)
 - Read, [466](#)
 - Set, [466](#)
 - SetFromDataElement, [466](#)
 - SetLength, [466](#)
 - SetNoSwap, [466](#)
 - SetValue, [467](#)
 - Type, [465](#)
 - Write, [467](#)
- gdcmm::Element< TVR, VM::VM1_n >, [467](#)
 - ~Element, [469](#)
 - Element, [469](#)
 - GetAsDataElement, [470](#)
 - GetLength, [470](#)
 - GetValue, [470](#)
 - GetValues, [470](#)
 - GetVM, [470](#)
 - GetVR, [470](#)
 - operator=, [471](#)
 - operator[], [471](#)
 - Print, [471](#)
 - Read, [471](#)
 - Set, [471](#)
 - SetArray, [472](#)
 - SetFromDataElement, [472](#)
 - SetLength, [472](#)
 - SetNoSwap, [472](#)
 - SetValue, [473](#)
 - Type, [469](#)
 - Write, [473](#)
 - WriteASCII, [473](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [474](#)
 - GetAsDataElement, [478](#)
 - GetLength, [478](#)
 - GetValue, [478](#)
 - GetValues, [478](#)
 - GetVM, [478](#)
 - GetVR, [478](#)
 - Internal, [480](#)
 - operator[], [478](#)
 - Parent, [477](#)
 - Print, [478](#)
 - Read, [478](#)
 - Set, [479](#)
 - SetFromDataElement, [479](#)
 - SetLength, [479](#)
 - SetNoSwap, [479](#)
 - SetValue, [479](#)
 - Type, [477](#)
 - Write, [479](#)
- gdcmm::Element< TVR, VM::VM2_n >, [480](#)
 - GetAsDataElement, [483](#)
 - GetLength, [483](#)
 - GetValue, [483](#)
 - GetValues, [483](#)
 - GetVM, [483](#)
 - GetVR, [484](#)
 - Internal, [485](#)
 - operator[], [484](#)
 - Parent, [483](#)
 - Print, [484](#)
 - Read, [484](#)
 - Set, [484](#)
 - SetFromDataElement, [484](#)
 - SetLength, [484](#)
 - SetNoSwap, [484](#)
 - SetValue, [485](#)
 - Type, [483](#)
 - Write, [485](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [485](#)
 - GetAsDataElement, [490](#)
 - GetLength, [490](#)
 - GetValue, [490](#)
 - GetValues, [490](#)
 - GetVM, [490](#)
 - GetVR, [490](#)
 - Internal, [492](#)
 - operator[], [490](#)
 - Parent, [489](#)

- Print, [490](#)
- Read, [490](#)
- Set, [491](#)
- SetFromDataElement, [491](#)
- SetLength, [491](#)
- SetNoSwap, [491](#)
- SetValue, [491](#)
- Type, [489](#)
- Write, [491](#)
- gdcmm::Element< TVR, VM::VM3_4 >, [492](#)
 - GetAsDataElement, [495](#)
 - GetLength, [495](#)
 - GetValue, [495](#)
 - GetValues, [495](#)
 - GetVM, [495](#)
 - GetVR, [496](#)
 - Internal, [497](#)
 - operator[], [496](#)
 - Parent, [495](#)
 - Print, [496](#)
 - Read, [496](#)
 - Set, [496](#)
 - SetFromDataElement, [496](#)
 - SetLength, [496](#)
 - SetNoSwap, [496](#)
 - SetValue, [497](#)
 - Type, [495](#)
 - Write, [497](#)
- gdcmm::Element< TVR, VM::VM3_n >, [497](#)
 - GetAsDataElement, [501](#)
 - GetLength, [501](#)
 - GetValue, [501](#)
 - GetValues, [501](#)
 - GetVM, [501](#)
 - GetVR, [502](#)
 - Internal, [503](#)
 - operator[], [502](#)
 - Parent, [501](#)
 - Print, [502](#)
 - Read, [502](#)
 - Set, [502](#)
 - SetFromDataElement, [502](#)
 - SetLength, [502](#)
 - SetNoSwap, [502](#)
 - SetValue, [503](#)
 - Type, [501](#)
 - Write, [503](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [503](#)
 - GetAsDataElement, [505](#)
 - GetLength, [505](#)
 - GetValue, [505](#)
 - GetValues, [505](#)
 - GetVM, [505](#)
 - GetVR, [505](#)
- Internal, [507](#)
- operator[], [506](#)
- Print, [506](#)
- Read, [506](#)
- Set, [506](#)
- SetFromDataElement, [506](#)
- SetNoSwap, [506](#)
- SetValue, [506](#)
- Type, [505](#)
- Write, [506](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [507](#)
 - GetAsDataElement, [510](#)
 - GetLength, [510](#)
 - GetValue, [510](#)
 - GetValues, [510](#)
 - GetVM, [510](#)
 - GetVR, [510](#)
 - Internal, [512](#)
 - operator[], [510](#)
 - Print, [511](#)
 - Read, [511](#)
 - Set, [511](#)
 - SetFromDataElement, [511](#)
 - SetNoSwap, [511](#)
 - SetValue, [511](#)
 - Type, [510](#)
 - Write, [511](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [512](#)
 - GetAsDataElement, [515](#)
 - GetLength, [515](#)
 - GetValue, [515](#)
 - GetValues, [515](#)
 - GetVM, [515](#)
 - GetVR, [515](#)
 - Internal, [517](#)
 - operator[], [515](#)
 - Print, [516](#)
 - Read, [516](#)
 - Set, [516](#)
 - SetFromDataElement, [516](#)
 - SetNoSwap, [516](#)
 - SetValue, [516](#)
 - Type, [515](#)
 - Write, [516](#)
- gdcmm::ElementDisableCombinations< TVR, TVM >, [517](#)
- gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >, [518](#)
- gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >, [519](#)
- gdcmm::EmptyMaskGenerator, [520](#)
 - ~EmptyMaskGenerator, [521](#)
 - EmptyMaskGenerator, [521](#)
 - Execute, [521](#)

- SetInputDirectory, 521
- SetOutputDirectory, 522
- SetSOPClassUIDMode, 522
- SOPClassUIDMode, 521
- UseGrayscaleSecondaryImageStorage, 521
- UseOriginalSOPClassUID, 521
- gdcm::EncapsulatedDocument, 522
 - EncapsulatedDocument, 523
- gdcm::EncodingImplementation< T >, 523
- gdcm::EncodingImplementation< VR::VRASCII >, 524
 - Read, 525
 - ReadComputeLength, 525
 - ReadNoSwap, 525
 - Write, 525, 526
- gdcm::EncodingImplementation< VR::VRBINARY >, 526
 - Read, 527
 - ReadComputeLength, 527
 - ReadNoSwap, 527
 - Write, 528
- gdcm::EndEvent, 528
- gdcm::EnumeratedValues, 529
 - EnumeratedValues, 530
- gdcm::EquipmentManufacturer, 530
 - AGFA, 531
 - Compute, 531
 - FUJI, 531
 - GEMS, 531
 - HITACHI, 531
 - KODAK, 531
 - MARCONI, 531
 - PMS, 531
 - SAMSUNG, 531
 - SIEMENS, 531
 - TOSHIBA, 531
 - Type, 531
 - TypeToString, 531
 - UIH, 531
 - UNKNOWN, 531
- gdcm::Event, 532
 - ~Event, 533
 - CheckEvent, 534
 - Event, 533
 - GetEventName, 534
 - MakeObject, 534
 - operator=, 534
 - Print, 534
- gdcm::Exception, 535
 - ~Exception, 537
 - Exception, 537
 - GetDescription, 537
 - what, 537
- gdcm::ExitEvent, 538
- gdcm::ExplicitDataElement, 539
 - GetLength, 542
 - Read, 542
 - ReadPreValue, 542
 - ReadValue, 542
 - ReadWithLength, 543
 - Write, 543
- gdcm::ExplicitImplicitDataElement, 543
 - GetLength, 546
 - Read, 546
 - ReadPreValue, 546
 - ReadValue, 546
 - ReadWithLength, 546
- gdcm::Fiducials, 547
 - Fiducials, 547
- gdcm::File, 548
 - ~File, 550
 - File, 550
 - GetDataSet, 550
 - GetHeader, 551
 - operator<<, 552
 - Read, 551
 - SetDataSet, 551
 - SetHeader, 552
 - Write, 552
- gdcm::FileAnonymizer, 553
 - ~FileAnonymizer, 555
 - Empty, 555
 - FileAnonymizer, 555
 - Remove, 555
 - Replace, 556
 - SetInputFileName, 556
 - SetOutputFileName, 556
 - Write, 557
- gdcm::FileChangeTransferSyntax, 557
 - ~FileChangeTransferSyntax, 559
 - Change, 560
 - FileChangeTransferSyntax, 559
 - GetCodec, 560
 - New, 560
 - SetInputFileName, 560
 - SetOutputFileName, 560
 - SetTransferSyntax, 561
- gdcm::FileDecompressLookupTable, 561
 - ~FileDecompressLookupTable, 563
 - Change, 563
 - FileDecompressLookupTable, 563
 - GetFile, 563
 - GetPixmap, 564
 - SetFile, 564
 - SetPixmap, 564
- gdcm::FileDerivation, 564
 - ~FileDerivation, 565
 - AddDerivationDescription, 566

- AddPurposeOfReferenceCodeSequence, 566
- AddReference, 566
- AddSourceImageSequence, 566
- Derive, 566
- FileDerivation, 565
- GetFile, 566, 567
- SetAppendDerivationHistory, 567
- SetDerivationCodeSequenceCodeValue, 567
- SetDerivationDescription, 567
- SetFile, 567
- SetPurposeOfReferenceCodeSequenceCodeValue, 568
- gdcm::FileExplicitFilter, 568
 - ~FileExplicitFilter, 569
 - Change, 569
 - ChangeFMI, 569
 - FileExplicitFilter, 569
 - GetFile, 570
 - ProcessDataSet, 570
 - SetChangePrivateTags, 570
 - SetFile, 570
 - SetRecomputeItemLength, 570
 - SetRecomputeSequenceLength, 571
 - SetUseVRUN, 571
- gdcm::FileMetaInformation, 571
 - ~FileMetaInformation, 575
 - AppendImplementationClassUID, 576
 - ComputeDataSetMediaStorageSOPClass, 576
 - ComputeDataSetTransferSyntax, 576
 - DataSetMS, 581
 - DataSetTS, 581
 - Default, 576
 - FileMetaInformation, 575, 576
 - FillFromDataSet, 576
 - GetDataSetTransferSyntax, 576
 - GetFileMetaInformationVersion, 577
 - GetFullLength, 577
 - GetGDCMImplementationClassUID, 577
 - GetGDCMImplementationVersionName, 577
 - GetGDCMSourceApplicationEntityTitle, 577
 - GetImplementationClassUID, 577
 - GetImplementationVersionName, 577
 - GetMediaStorage, 577
 - GetMediaStorageAsString, 578
 - GetMetaInformationTS, 578
 - GetPreamble, 578
 - GetSourceApplicationEntityTitle, 578
 - Insert, 578
 - IsValid, 578
 - MetaInformationTS, 581
 - operator<<, 580
 - operator=, 578
 - Read, 579
 - ReadCompat, 579
 - ReadCompatInternal, 579
 - Replace, 579
 - SetDataSetTransferSyntax, 579
 - SetImplementationClassUID, 579
 - SetImplementationVersionName, 580
 - SetPreamble, 580
 - SetSourceApplicationEntityTitle, 580
 - Write, 580
- gdcm::Filename, 581
 - EndWith, 582
 - Filename, 582
 - GetExtension, 582
 - GetFileName, 582
 - GetName, 582
 - GetPath, 583
 - IsEmpty, 583
 - IsIdentical, 583
 - Join, 583
 - operator const char *, 583
 - ToUnixSlashes, 583
 - ToWindowsSlashes, 584
- gdcm::FileNameEvent, 584
 - ~FileNameEvent, 586
 - CheckEvent, 587
 - FileNameEvent, 586
 - GetEventName, 587
 - GetFileName, 587
 - MakeObject, 587
 - operator=, 587
 - Self, 586
 - SetFileName, 587
 - Superclass, 586
- gdcm::FilenameGenerator, 588
 - ~FilenameGenerator, 589
 - FilenameGenerator, 589
 - FilenamesType, 589
 - FilenameType, 589
 - Generate, 589
 - GetFilename, 589
 - GetFilenames, 590
 - GetNumberOfFilenames, 590
 - GetPattern, 590
 - GetPrefix, 590
 - SetNumberOfFilenames, 590
 - SetPattern, 590
 - SetPrefix, 591
 - SizeType, 589
- gdcm::FileSet, 591
 - AddFile, 592
 - FileSet, 592
 - FilesType, 592
 - FileType, 592
 - GetFiles, 592
 - operator<<, 593

- SetFiles, 593
- gdcm::FileStreamer, 593
 - ~FileStreamer, 596
 - AppendToDataElement, 596
 - AppendToGroupDataElement, 596
 - CheckDataElement, 596
 - CheckTemplateFileName, 596
 - FileStreamer, 596
 - New, 597
 - ReserveDataElement, 597
 - ReserveGroupDataElement, 597
 - SetOutputFileName, 597
 - SetTemplateFileName, 597
 - StartDataElement, 598
 - StartGroupDataElement, 598
 - StopDataElement, 598
 - StopGroupDataElement, 598
- gdcm::FileWithName, 599
 - filename, 601
 - FileWithName, 601
- gdcm::FindPatientRootQuery, 602
 - FindPatientRootQuery, 604
 - GetAbstractSyntaxUID, 604
 - GetTagListByLevel, 604
 - InitializeDataSet, 605
 - QueryFactory, 605
 - ValidateQuery, 605
- gdcm::FindStudyRootQuery, 606
 - FindStudyRootQuery, 608
 - GetAbstractSyntaxUID, 608
 - GetTagListByLevel, 608
 - InitializeDataSet, 609
 - QueryFactory, 609
 - ValidateQuery, 609
- gdcm::Fragment, 610
 - ComputeLength, 613
 - Fragment, 613
 - GetLength, 613
 - operator<<, 614
 - Read, 613
 - ReadBacktrack, 613
 - ReadPreValue, 613
 - ReadValue, 614
 - Write, 614
- gdcm::Global, 615
 - ~Global, 616
 - Append, 616
 - GetDefs, 616
 - GetDicts, 616, 617
 - GetInstance, 617
 - Global, 616
 - LoadResourcesFiles, 617
 - Locate, 617
 - operator<<, 618
 - operator=, 618
 - Prepend, 618
- gdcm::GroupDict, 618
 - ~GroupDict, 619
 - Add, 620
 - GetAbbreviation, 620
 - GetName, 620
 - GroupDict, 619
 - GroupStringVector, 619
 - Insert, 620
 - operator<<, 621
 - Size, 620
- gdcm::IconImageFilter, 621
 - ~IconImageFilter, 622
 - Extract, 623
 - ExtractIconImages, 623
 - ExtractVeproIconImages, 623
 - GetFile, 623
 - GetIconImage, 623
 - GetNumberOfIconImages, 623
 - IconImageFilter, 622
 - SetFile, 624
- gdcm::IconImageGenerator, 624
 - ~IconImageGenerator, 625
 - AutoPixelMinMax, 625
 - ConvertRGBToPaletteColor, 625
 - Generate, 626
 - GetIconImage, 626
 - GetPixmap, 626
 - IconImageGenerator, 625
 - SetOutputDimensions, 626
 - SetOutsideValuePixel, 626
 - SetPixelMinMax, 627
 - SetPixmap, 627
- gdcm::ignore_char, 627
 - ignore_char, 628
 - m_char, 628
- gdcm::Image, 628
 - ~Image, 633
 - GetDirectionCosines, 634
 - GetIntercept, 634
 - GetOrigin, 634
 - GetSlope, 634
 - GetSpacing, 634
 - Image, 633
 - Print, 635
 - SetDirectionCosines, 635
 - SetIntercept, 635
 - SetOrigin, 635, 636
 - SetSlope, 636
 - SetSpacing, 636
- gdcm::ImageApplyLookupTable, 637
 - ~ImageApplyLookupTable, 639
 - Apply, 640

- ImageApplyLookupTable, 639
- SetRGB8, 640
- gdcmm::ImageChangePhotometricInterpretation, 640
 - ~ImageChangePhotometricInterpretation, 643
 - Change, 643
 - ChangeMonochrome, 643
 - ChangeRGB2YBR, 643
 - ChangeYBR2RGB, 643
 - GetPhotometricInterpretation, 643
 - ImageChangePhotometricInterpretation, 643
 - RGB2YBR, 643
 - SetPhotometricInterpretation, 644
 - YBR2RGB, 644
- gdcmm::ImageChangePlanarConfiguration, 645
 - ~ImageChangePlanarConfiguration, 648
 - Change, 648
 - GetPlanarConfiguration, 648
 - ImageChangePlanarConfiguration, 648
 - RGBPixelsToRGBPlanes, 648
 - RGBPlanesToRGBPixels, 648
 - SetPlanarConfiguration, 649
- gdcmm::ImageChangeTransferSyntax, 649
 - ~ImageChangeTransferSyntax, 653
 - Change, 653
 - GetTransferSyntax, 653
 - ImageChangeTransferSyntax, 653
 - SetCompressIconImage, 654
 - SetForce, 654
 - SetTransferSyntax, 654
 - SetUserCodec, 654
 - TryJPEG2000Codec, 655
 - TryJPEGCodec, 655
 - TryJPEGLSCodec, 655
 - TryRAWCodec, 655
 - TryRLECodec, 655
- gdcmm::ImageCodec, 656
 - ~ImageCodec, 659
 - AppendFrameEncode, 659
 - AppendRowEncode, 659
 - CanCode, 660
 - CanDecode, 660
 - CleanupUnusedBits, 660
 - Clone, 660
 - Decode, 660
 - DecodeByStreams, 661
 - Dimensions, 667
 - DoByteSwap, 661
 - DoInvertMonochrome, 661
 - DoOverlayCleanup, 661
 - DoPaddedCompositePixelCode, 661
 - DoPlanarConfiguration, 661
 - DoSimpleCopy, 662
 - DoYBR, 662
 - DoYBRFull422, 662
- FileChangeTransferSyntax, 666
- GetDimensions, 662
- GetHeaderInfo, 662
- GetLossyFlag, 662
- GetLUT, 662
- GetNeedByteSwap, 663
- GetNumberOfDimensions, 663
- GetPhotometricInterpretation, 663
- GetPixelFormat, 663
- GetPlanarConfiguration, 663
- ImageChangePhotometricInterpretation, 666
- ImageCodec, 659
- IsFrameEncoder, 663
- IsLossy, 664
- IsRowEncoder, 664
- IsValid, 664
- LossyFlag, 667
- LUT, 667
- LUTPtr, 659
- NeedByteSwap, 667
- NeedOverlayCleanup, 667
- NumberOfDimensions, 667
- PF, 667
- PI, 668
- PlanarConfiguration, 668
- RequestPaddedCompositePixelCode, 668
- RequestPlanarConfiguration, 668
- SetDimensions, 664
- SetLossyFlag, 664
- SetLUT, 664
- SetNeedByteSwap, 665
- SetNeedOverlayCleanup, 665
- SetNumberOfDimensions, 665
- SetPhotometricInterpretation, 665
- SetPixelFormat, 665
- SetPlanarConfiguration, 666
- StartEncode, 666
- StopEncode, 666
- gdcmm::ImageConverter, 668
 - ~ImageConverter, 669
 - Convert, 669
 - GetOutput, 669
 - ImageConverter, 669
 - SetInput, 669
- gdcmm::ImageFragmentSplitter, 670
 - ~ImageFragmentSplitter, 672
 - GetFragmentSizeMax, 673
 - ImageFragmentSplitter, 672
 - SetForce, 673
 - SetFragmentSizeMax, 673
 - Split, 673
- gdcmm::ImageHelper, 673
 - ComputeMediaStorageFromModality, 675
 - ComputeSpacingFromImagePositionPatient, 675

- GetDimensionsValue, 675
- GetDirectionCosinesFromDataSet, 675
- GetDirectionCosinesValue, 676
- GetForcePixelSpacing, 676
- GetForceRescaleInterceptSlope, 676
- GetLUT, 676
- GetOriginValue, 676
- GetPhotometricInterpretationValue, 676
- GetPixelFormatValue, 676
- GetPlanarConfigurationValue, 677
- GetPMSRescaleInterceptSlope, 677
- GetPointerFromElement, 677
- GetRealWorldValueMappingContent, 677
- GetRescaleInterceptSlopeValue, 677
- GetSecondaryCaptureImagePlaneModule, 677
- GetSpacingTagFromMediaStorage, 678
- GetSpacingValue, 678
- GetZSpacingTagFromMediaStorage, 678
- SetDimensionsValue, 678
- SetDirectionCosinesValue, 678
- SetForcePixelSpacing, 678
- SetForceRescaleInterceptSlope, 678
- SetOriginValue, 679
- SetPMSRescaleInterceptSlope, 679
- SetRescaleInterceptSlopeValue, 679
- SetSecondaryCaptureImagePlaneModule, 679
- SetSpacingValue, 679
- gdcm::ImageReader, 680
 - ~ImageReader, 683
 - GetImage, 683
 - ImageReader, 683
 - Read, 683
 - ReadACRNEMAIImage, 684
 - ReadImage, 684
- gdcm::ImageRegionReader, 685
 - ~ImageRegionReader, 688
 - ComputeBufferLength, 689
 - GetRegion, 689
 - ImageRegionReader, 688
 - Read, 689
 - ReadInformation, 689
 - ReadIntoBuffer, 689
 - SetRegion, 690
- gdcm::ImageToImageFilter, 690
 - ~ImageToImageFilter, 692
 - GetInput, 692
 - GetOutput, 692
 - ImageToImageFilter, 692
- gdcm::ImageWriter, 693
 - ~ImageWriter, 696
 - ComputeTargetMediaStorage, 696
 - GetImage, 696, 697
 - ImageWriter, 696
 - Write, 697
- gdcm::ImplicitDataElement, 701
 - GetLength, 704
 - Read, 704
 - ReadPreValue, 704
 - ReadValue, 704
 - ReadValueWithLength, 704
 - ReadWithLength, 704
 - Write, 704
- gdcm::InitializeEvent, 705
- gdcm::IOD, 706
 - AddIODEntry, 708
 - Clear, 708
 - GetIODEntry, 708
 - GetNumberOfIODs, 708
 - GetTypeFromTag, 708
 - IOD, 707
 - MapIODEntry, 707
 - operator<<, 708
 - SizeType, 707
- gdcm::IODEntry, 709
 - GetIE, 710
 - GetName, 710
 - GetRef, 710
 - GetUsage, 710
 - GetUsageType, 710
 - IODEntry, 710
 - operator<<, 711
 - SetIE, 710
 - SetName, 710
 - SetRef, 711
 - SetUsage, 711
- gdcm::IODs, 711
 - AddIOD, 713
 - Begin, 713
 - Clear, 713
 - End, 713
 - GetIOD, 713
 - IODMapType, 712
 - IODMapTypeConstIterator, 712
 - IODName, 712
 - IODs, 713
 - operator<<, 714
- gdcm::IPPSorter, 714
 - ComputeZSpacing, 719
 - DirCosTolerance, 719
 - DropDuplicatePositions, 719
 - GetDirectionCosinesTolerance, 717
 - GetZSpacing, 717
 - GetZSpacingTolerance, 717
 - IPPSorter, 717
 - SetComputeZSpacing, 717
 - SetDirectionCosinesTolerance, 718
 - SetDropDuplicatePositions, 718
 - SetZSpacingTolerance, 718

- Sort, [718](#)
- ZSpacing, [719](#)
- ZTolerance, [719](#)
- gdcm::Item, [720](#)
 - Clear, [724](#)
 - FindDataElement, [724](#)
 - GetDataElement, [724](#)
 - GetLength, [724](#)
 - GetNestedDataSet, [724](#)
 - InsertDataElement, [725](#)
 - Item, [723](#)
 - operator<<, [726](#)
 - Read, [725](#)
 - SetNestedDataSet, [725](#)
 - Write, [725](#)
- gdcm::IterationEvent, [726](#)
- gdcm::JPEG12Codec, [727](#)
 - ~JPEG12Codec, [731](#)
 - DecodeByStreams, [731](#)
 - EncodeBuffer, [731](#)
 - GetHeaderInfo, [732](#)
 - InternalCode, [732](#)
 - IsStateSuspension, [732](#)
 - JPEG12Codec, [731](#)
- gdcm::JPEG16Codec, [733](#)
 - ~JPEG16Codec, [736](#)
 - DecodeByStreams, [736](#)
 - EncodeBuffer, [736](#)
 - GetHeaderInfo, [737](#)
 - InternalCode, [737](#)
 - IsStateSuspension, [737](#)
 - JPEG16Codec, [736](#)
- gdcm::JPEG2000Codec, [738](#)
 - ~JPEG2000Codec, [741](#)
 - AppendFrameEncode, [741](#)
 - AppendRowEncode, [741](#)
 - Bitmap, [745](#)
 - CanCode, [742](#)
 - CanDecode, [742](#)
 - Clone, [742](#)
 - Code, [742](#)
 - Decode, [742](#)
 - DecodeByStreams, [743](#)
 - DecodeExtent, [743](#)
 - GetHeaderInfo, [743](#)
 - GetQuality, [743](#)
 - GetRate, [743](#)
 - ImageRegionReader, [745](#)
 - IsFrameEncoder, [744](#)
 - IsRowEncoder, [744](#)
 - JPEG2000Codec, [741](#)
 - SetMCT, [744](#)
 - SetNumberOfResolutions, [744](#)
 - SetNumberOfThreadsForDecompression, [744](#)
 - SetQuality, [744](#)
 - SetRate, [744](#)
 - SetReversible, [745](#)
 - SetTileSize, [745](#)
 - StartEncode, [745](#)
 - StopEncode, [745](#)
- gdcm::JPEG8Codec, [746](#)
 - ~JPEG8Codec, [750](#)
 - DecodeByStreams, [750](#)
 - EncodeBuffer, [750](#)
 - GetHeaderInfo, [750](#)
 - InternalCode, [750](#)
 - IsStateSuspension, [750](#)
 - JPEG8Codec, [750](#)
- gdcm::JPEGCodec, [751](#)
 - ~JPEGCodec, [755](#)
 - AppendFrameEncode, [755](#)
 - AppendRowEncode, [755](#)
 - BitSample, [760](#)
 - CanCode, [755](#)
 - CanDecode, [755](#)
 - Clone, [756](#)
 - Code, [756](#)
 - ComputeOffsetTable, [756](#)
 - Decode, [756](#)
 - DecodeByStreams, [756](#)
 - DecodeExtent, [757](#)
 - EncodeBuffer, [757](#)
 - GetHeaderInfo, [757](#)
 - GetLossless, [757](#)
 - GetQuality, [757](#)
 - ImageRegionReader, [759](#)
 - IsFrameEncoder, [758](#)
 - IsRowEncoder, [758](#)
 - IsStateSuspension, [758](#)
 - IsValid, [758](#)
 - JPEGCodec, [755](#)
 - Quality, [760](#)
 - SetBitSample, [758](#)
 - SetLossless, [758](#)
 - SetPixelFormat, [758](#)
 - SetQuality, [759](#)
 - StartEncode, [759](#)
 - StopEncode, [759](#)
- gdcm::JPEGLSCodec, [760](#)
 - ~JPEGLSCodec, [763](#)
 - AppendFrameEncode, [764](#)
 - AppendRowEncode, [764](#)
 - CanCode, [764](#)
 - CanDecode, [764](#)
 - Clone, [764](#)
 - Code, [765](#)
 - Decode, [765](#)
 - DecodeExtent, [765](#)

- GetBufferLength, [765](#)
- GetHeaderInfo, [766](#)
- GetLossless, [766](#)
- ImageRegionReader, [767](#)
- IsFrameEncoder, [766](#)
- IsRowEncoder, [766](#)
- JPEGLSCodec, [763](#)
- SetBufferLength, [766](#)
- SetLossless, [766](#)
- SetLossyError, [766](#)
- StartEncode, [766](#)
- StopEncode, [767](#)
- gdcmm::JSON, [767](#)
 - ~JSON, [768](#)
 - Code, [768](#)
 - Decode, [768](#)
 - GetPrettyPrint, [768](#)
 - JSON, [768](#)
 - PrettyPrintOff, [768](#)
 - PrettyPrintOn, [769](#)
 - SetPrettyPrint, [769](#)
- gdcmm::KAKADUCodec, [769](#)
 - ~KAKADUCodec, [772](#)
 - CanCode, [772](#)
 - CanDecode, [772](#)
 - Clone, [772](#)
 - Code, [772](#)
 - Decode, [773](#)
 - KAKADUCodec, [772](#)
- gdcmm::LO, [773](#)
 - const_iterator, [775](#)
 - const_reference, [775](#)
 - const_reverse_iterator, [775](#)
 - difference_type, [775](#)
 - IsValid, [776](#)
 - iterator, [775](#)
 - LO, [776](#)
 - pointer, [775](#)
 - reference, [775](#)
 - reverse_iterator, [775](#)
 - size_type, [775](#)
 - Superclass, [775](#)
 - value_type, [776](#)
- gdcmm::LookupTable, [777](#)
 - ~LookupTable, [779](#)
 - Allocate, [780](#)
 - BitSample, [784](#)
 - BLUE, [779](#)
 - Clear, [780](#)
 - Decode, [780](#)
 - Decode8, [780](#)
 - GetBitSample, [781](#)
 - GetBufferAsRGBA, [781](#)
 - GetLUT, [781](#)
 - GetLUTDescriptor, [781](#)
 - GetLUTLength, [781](#)
 - GetPointer, [781](#)
 - GRAY, [779](#)
 - GREEN, [779](#)
 - IncompleteLUT, [784](#)
 - InitializeBlueLUT, [782](#)
 - Initialized, [782](#)
 - InitializeGreenLUT, [782](#)
 - InitializeLUT, [782](#)
 - InitializeRedLUT, [782](#)
 - Internal, [784](#)
 - IsRGB8, [782](#)
 - LookupTable, [779](#), [780](#)
 - LookupTableType, [779](#)
 - Print, [783](#)
 - RED, [779](#)
 - SetBlueLUT, [783](#)
 - SetGreenLUT, [783](#)
 - SetLUT, [783](#)
 - SetRedLUT, [783](#)
 - UNKNOWN, [779](#)
 - WriteBufferAsRGBA, [783](#)
- gdcmm::Macro, [787](#)
 - AddMacroEntry, [788](#)
 - ArrayIncludeMacrosType, [787](#)
 - Clear, [788](#)
 - FindMacroEntry, [788](#)
 - GetMacroEntry, [788](#)
 - GetName, [788](#)
 - Macro, [788](#)
 - MapModuleEntry, [787](#)
 - operator<<, [789](#)
 - SetName, [788](#)
 - Verify, [789](#)
- gdcmm::Macros, [789](#)
 - AddMacro, [790](#)
 - Clear, [790](#)
 - GetMacro, [791](#)
 - IsEmpty, [791](#)
 - Macros, [790](#)
 - ModuleMapType, [790](#)
 - operator<<, [791](#)
- gdcmm::MD5, [793](#)
 - Compute, [793](#)
 - ComputeFile, [793](#)
- gdcmm::MEC_MR3, [794](#)
 - GetCanonMECMR3Tag, [794](#)
 - GetPMTFInformationDataTag, [794](#)
 - GetToshibaMECMR3Tag, [794](#)
 - Print, [795](#)
- gdcmm::MediaStorage, [795](#)
 - AmbulatoryECGWaveformStorage, [799](#)
 - Audio, [802](#)

- BasicTextSR, 800
- BasicVoiceAudioWaveformStorage, 799
- BreastProjectionXRayImageStorageForPresentation, 801
- BreastProjectionXRayImageStorageForProcessing, 801
- BreastTomosynthesisImageStorage, 801
- CardiacElectrophysiologyWaveformStorage, 799
- ComprehensiveSR, 800
- ComputedRadiographyImageStorage, 799
- CSANonImageStorage, 800
- CTImageStorage, 799
- DetachedPatientManagementSOPClass, 800
- DetachedStudyManagementSOPClass, 800
- DetachedVisitManagementSOPClass, 800
- DigitalIntraoralXrayImageStorageForPresentation, 799
- DigitalIntraoralXRayImageStorageForProcessing, 799
- DigitalMammographyImageStorageForPresentation, 799
- DigitalMammographyImageStorageForProcessing, 799
- DigitalXRayImageStorageForPresentation, 799
- DigitalXRayImageStorageForProcessing, 799
- EncapsulatedCDASStorage, 800
- EncapsulatedPDFStorage, 800
- EnhancedCTImageStorage, 799
- EnhancedMRColorImageStorage, 801
- EnhancedMRImageStorage, 799
- EnhancedPETImageStorage, 801
- EnhancedSR, 800
- EnhancedUSVolumeStorage, 801
- EnhancedXAImageStorage, 800
- FujiPrivateCRImageStorage, 801
- FujiPrivateMammoCRImageStorage, 801
- GeneralECGWaveformStorage, 799
- GeneralElectricMagneticResonanceImageStorage, 800
- GEPrivate3DModelStorage, 800
- GetModality, 802
- GetModalityDimension, 802
- GetMSString, 802
- GetMSType, 802
- GetNumberOfModality, 803
- GetNumberOfMSString, 803
- GetNumberOfMSType, 803
- GetString, 803
- GrayscaleSoftcopyPresentationStateStorageSOPClass, 800
- GuessFromModality, 803
- HangingProtocolStorage, 800
- HardcopyColorImageStorage, 801
- HardcopyGrayscaleImageStorage, 800
- HemodynamicWaveformStorage, 799
- IsImage, 803
- IsUndefined, 804
- IVOCTForPresentation, 801
- IVOCTForProcessing, 801
- KeyObjectSelectionDocument, 800
- LeadECGWaveformStorage, 799
- LegacyConvertedEnhancedCTImageStorage, 801
- LegacyConvertedEnhancedMRImageStorage, 801
- LegacyConvertedEnhancedPETImageStorage, 801
- MammographyCADSR, 800
- MediaStorage, 802
- MediaStorageDirectoryStorage, 799
- ModalityPerformedProcedureStepSOPClass, 800
- MRImageStorage, 799
- MRSpectroscopyStorage, 799
- MS_END, 801
- MSType, 799
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, 799
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, 799
- MultiframeSingleBitSecondaryCaptureImageStorage, 799
- MultiframeTrueColorSecondaryCaptureImageStorage, 799
- NoObject, 801
- NuclearMedicineImageStorage, 800
- NuclearMedicineImageStorageRetired, 799
- ObjectEnd, 802
- ObjectType, 801
- operator MSType, 804
- operator<<, 805
- OphthalmicPhotography16BitImageStorage, 801
- OphthalmicPhotography8BitImageStorage, 801
- OphthalmicTomographyImageStorage, 801
- PDF, 802
- PETImageStorage, 800
- Philips3D, 800
- PhilipsPrivateMRSyntheticImageStorage, 800
- RawDataStorage, 800
- RTDoseStorage, 800
- RTImageStorage, 800
- RTIonBeamsTreatmentRecordStorage, 801
- RTIonPlanStorage, 800
- RTPlanStorage, 800
- RTStructureSetStorage, 800
- RTTreatmentSummaryRecordStorage, 801
- SecondaryCaptureImageStorage, 799
- Segmentation, 802

- SegmentationStorage, 800
- SetFromDataSet, 804
- SetFromFile, 804
- SetFromHeader, 804
- SetFromModality, 805
- SetFromSourceImageSequence, 805
- SpacialFiducialsStorage, 800
- SpacialRegistrationStorage, 800
- StandaloneCurveStorage, 799
- StandaloneModalityLUTStorage, 799
- StandaloneOverlayStorage, 799
- StandaloneVOILUTStorage, 799
- StudyComponentManagementSOPClass, 800
- SurfaceSegmentationStorage, 801
- ToshibaPrivateDataStorage, 800
- UltrasoundImageStorage, 799
- UltrasoundImageStorageRetired, 799
- UltrasoundMultiFrameImageStorage, 799
- UltrasoundMultiFrameImageStorageRetired, 799
- URI, 802
- Video, 802
- VideoEndoscopicImageStorage, 800
- VideoMicroscopicImageStorage, 801
- VideoPhotographicImageStorage, 801
- VLEndoscopicImageStorage, 801
- VLMicroscopicImageStorage, 801
- VLPhotographicImageStorage, 800
- VLWholeSlideMicroscopyImageStorage, 801
- Waveform, 802
- XRay3DAngiographicImageStorage, 800
- XRay3DCraniofacialImageStorage, 801
- XRayAngiographicBiPlaneImageStorageRetired, 800
- XRayAngiographicImageStorage, 800
- XRayRadiationDoseSR, 801
- XRayRadiofluoroscopicImageStorage, 800
- gdcM::MemberCommand< T >, 805
 - ~MemberCommand, 809
 - Execute, 810
 - m_ConstMemberFunction, 811
 - m_MemberFunction, 811
 - m_This, 811
 - MemberCommand, 809
 - New, 810
 - operator=, 810
 - Self, 809
 - SetCallbackFunction, 810, 811
 - TConstMemberFunctionPointer, 809
 - TMemberFunctionPointer, 809
- gdcM::MeshPrimitive, 812
 - ~MeshPrimitive, 815
 - AddPrimitiveData, 815
 - EDGE, 815
 - FACET, 815
 - GetMPType, 815
 - GetMPTypeString, 815
 - GetNumberOfPrimitivesData, 815
 - GetPrimitiveData, 816
 - GetPrimitivesData, 816
 - GetPrimitiveType, 816
 - LINE, 815
 - MeshPrimitive, 815
 - MPType, 814
 - MPType_END, 815
 - PrimitiveData, 817
 - PrimitivesData, 814
 - PrimitiveType, 817
 - SetPrimitiveData, 816
 - SetPrimitivesData, 817
 - SetPrimitiveType, 817
 - TRIANGLE, 815
 - TRIANGLE_FAN, 815
 - TRIANGLE_STRIP, 815
 - VERTEX, 815
- gdcM::ModalityPerformedProcedureStepCreateQuery, 817
 - GetAbstractSyntaxUID, 820
 - GetRequiredDataSet, 820
 - ModalityPerformedProcedureStepCreateQuery, 820
 - QueryFactory, 821
 - ValidateQuery, 820
- gdcM::ModalityPerformedProcedureStepSetQuery, 821
 - GetAbstractSyntaxUID, 824
 - GetRequiredDataSet, 824
 - ModalityPerformedProcedureStepSetQuery, 823
 - QueryFactory, 824
 - ValidateQuery, 824
- gdcM::ModifiedEvent, 825
- gdcM::Module, 826
 - AddMacro, 827
 - AddModuleEntry, 827
 - ArrayIncludeMacrosType, 827
 - Clear, 828
 - FindModuleEntryInMacros, 828
 - GetModuleEntryInMacros, 828
 - GetName, 828
 - MapModuleEntry, 827
 - Module, 827
 - operator<<, 829
 - SetName, 828
 - Verify, 828
- gdcM::ModuleEntry, 829
 - ~ModuleEntry, 831
 - DataElementType, 833
 - Description, 831

- DescriptionField, [833](#)
- GetDescription, [832](#)
- GetName, [832](#)
- GetType, [832](#)
- ModuleEntry, [831](#)
- Name, [833](#)
- operator<<, [833](#)
- SetDescription, [832](#)
- SetName, [832](#)
- SetType, [832](#)
- gdcmm::Modules, [833](#)
 - AddModule, [835](#)
 - Clear, [835](#)
 - GetModule, [835](#)
 - IsEmpty, [835](#)
 - ModuleMapType, [834](#)
 - Modules, [834](#)
 - operator<<, [835](#)
- gdcmm::MovePatientRootQuery, [836](#)
 - GetAbstractSyntaxUID, [838](#)
 - GetTagListByLevel, [838](#)
 - InitializeDataSet, [839](#)
 - MovePatientRootQuery, [838](#)
 - QueryFactory, [839](#)
 - ValidateQuery, [839](#)
- gdcmm::MoveStudyRootQuery, [840](#)
 - GetAbstractSyntaxUID, [842](#)
 - GetTagListByLevel, [842](#)
 - InitializeDataSet, [843](#)
 - MoveStudyRootQuery, [842](#)
 - QueryFactory, [843](#)
 - ValidateQuery, [843](#)
- gdcmm::MrProtocol, [844](#)
 - ~MrProtocol, [844](#)
 - FindMrProtocolByName, [845](#)
 - GetMrProtocolByName, [845](#)
 - GetSliceArray, [845](#)
 - GetVersion, [845](#)
 - Load, [845](#)
 - MrProtocol, [844](#)
 - operator<<, [846](#)
 - Print, [845](#)
- gdcmm::MrProtocol::Slice, [1134](#)
 - Normal, [1135](#)
 - Position, [1135](#)
- gdcmm::MrProtocol::SliceArray, [1135](#)
 - Slices, [1136](#)
- gdcmm::MrProtocol::Vector3, [1418](#)
 - dCor, [1418](#)
 - dSag, [1418](#)
 - dTra, [1418](#)
- gdcmm::NestedModuleEntries, [855](#)
 - AddModuleEntry, [857](#)
 - GetModuleEntry, [857](#)
 - GetNumberOfModuleEntries, [857](#)
 - NestedModuleEntries, [857](#)
 - operator<<, [858](#)
 - SizeType, [857](#)
- gdcmm::network, [102](#)
 - cMaxEventID, [108](#)
 - cMaxStateID, [108](#)
 - eAABORTPDUPReceivedOpen, [107](#)
 - eAABORTRequest, [107](#)
 - eAASSOCIATE_RQPDUPreceived, [107](#)
 - eAASSOCIATERequestLocalUser, [107](#)
 - eAASSOCIATEResponseAccept, [107](#)
 - eAASSOCIATEResponseReject, [107](#)
 - eARELEASE_RPPDUPReceived, [107](#)
 - eARELEASE_RQPDUPReceivedOpen, [107](#)
 - eARELEASERequest, [107](#)
 - eARELEASEResponse, [107](#)
 - eARTIMTimerExpired, [107](#)
 - eASSOCIATE_ACPDUPreceived, [107](#)
 - eASSOCIATE_RJPDUPreceived, [107](#)
 - eEventDoesNotExist, [107](#)
 - EEventID, [107](#)
 - ePDATArequest, [107](#)
 - ePDATATFPDUP, [107](#)
 - eSta10ReleaseCollisionAc, [108](#)
 - eSta11ReleaseCollisionRq, [108](#)
 - eSta12ReleaseCollisionAcLocal, [108](#)
 - eSta13AwaitingClose, [108](#)
 - eSta1Idle, [108](#)
 - eSta2Open, [108](#)
 - eSta3WaitLocalAssoc, [108](#)
 - eSta4LocalAssocDone, [108](#)
 - eSta5WaitRemoteAssoc, [108](#)
 - eSta6TransferReady, [108](#)
 - eSta7WaitRelease, [108](#)
 - eSta8WaitLocalRelease, [108](#)
 - eSta9ReleaseCollisionRqLocal, [108](#)
 - eStaDoesNotExist, [107](#)
 - EStateID, [107](#)
 - eTransportConnConfirmLocal, [107](#)
 - eTransportConnectionClosed, [107](#)
 - eTransportConnIndicLocal, [107](#)
 - eUnrecognizedPDUPReceived, [107](#)
 - GetStateIndex, [108](#)
- gdcmm::network::AAabortPDUP, [113](#)
 - AAabortPDUP, [114](#)
 - IsLastFragment, [114](#)
 - Print, [114](#)
 - Read, [114](#)
 - SetReason, [115](#)
 - SetSource, [115](#)
 - Size, [115](#)
 - Write, [115](#)
- gdcmm::network::AAAssociateACPDUP, [116](#)

- AAAssociateACPDU, [117](#)
- AAAssociateRQPDU, [119](#)
- AddPresentationContextAC, [118](#)
- GetNumberOfPresentationContextAC, [118](#)
- GetPresentationContextAC, [118](#)
- GetUserInformation, [118](#)
- InitFromRQ, [118](#)
- IsLastFragment, [118](#)
- Print, [118](#)
- Read, [118](#)
- SetCalledAETitle, [119](#)
- SetCallingAETitle, [119](#)
- Size, [119](#)
- SizeType, [117](#)
- Write, [119](#)
- gdcmm::network::AAAssociateRJPDU, [120](#)
 - AAAssociateRJPDU, [121](#)
 - IsLastFragment, [121](#)
 - Print, [121](#)
 - Read, [121](#)
 - Size, [121](#)
 - Write, [121](#)
- gdcmm::network::AAAssociateRQPDU, [122](#)
 - AAAssociateACPDU, [127](#)
 - AAAssociateRQPDU, [124](#)
 - AddPresentationContext, [124](#)
 - GetCalledAETitle, [124](#)
 - GetCallingAETitle, [124](#)
 - GetNumberOfPresentationContext, [125](#)
 - GetPresentationContext, [125](#)
 - GetPresentationContextByAbstractSyntax, [125](#)
 - GetPresentationContextByID, [125](#)
 - GetPresentationContexts, [125](#)
 - GetReserved43_74, [125](#)
 - GetUserInformation, [125](#)
 - IsAETitleValid, [125](#)
 - IsLastFragment, [126](#)
 - PresentationContextArrayType, [124](#)
 - Print, [126](#)
 - Read, [126](#)
 - SetCalledAETitle, [126](#)
 - SetCallingAETitle, [126](#)
 - SetUserInformation, [126](#)
 - Size, [127](#)
 - SizeType, [124](#)
 - Write, [127](#)
- gdcmm::network::AbstractSyntax, [129](#)
 - AbstractSyntax, [129](#)
 - GetAsDataElement, [130](#)
 - GetName, [130](#)
 - operator==, [130](#)
 - Print, [130](#)
 - Read, [130](#)
 - SetName, [130](#)
 - SetNameFromUID, [130](#)
 - Size, [130](#)
 - Write, [130](#)
- gdcmm::network::ApplicationContext, [145](#)
 - ApplicationContext, [146](#)
 - GetName, [146](#)
 - Print, [146](#)
 - Read, [146](#)
 - SetName, [146](#)
 - Size, [146](#)
 - Write, [146](#)
- gdcmm::network::AReleaseRPPDU, [149](#)
 - AReleaseRPPDU, [150](#)
 - IsLastFragment, [150](#)
 - Print, [150](#)
 - Read, [151](#)
 - Size, [151](#)
 - Write, [151](#)
- gdcmm::network::AReleaseRQPDU, [151](#)
 - AReleaseRQPDU, [153](#)
 - IsLastFragment, [153](#)
 - Print, [153](#)
 - Read, [153](#)
 - Size, [153](#)
 - Write, [153](#)
- gdcmm::network::ARTIMTimer, [154](#)
 - ARTIMTimer, [154](#)
 - GetElapsedTime, [154](#)
 - GetHasExpired, [154](#)
 - GetTimeout, [155](#)
 - SetTimeout, [155](#)
 - Start, [155](#)
 - Stop, [155](#)
- gdcmm::network::AsynchronousOperationsWindowSub, [157](#)
 - AsynchronousOperationsWindowSub, [157](#)
 - Print, [158](#)
 - Read, [158](#)
 - Size, [158](#)
 - Write, [158](#)
- gdcmm::network::BaseCompositeMessage, [224](#)
 - ~BaseCompositeMessage, [226](#)
 - ConstructPDV, [226](#)
- gdcmm::network::BaseNormalizedMessage, [226](#)
 - ~BaseNormalizedMessage, [228](#)
 - ConstructPDV, [228](#)
- gdcmm::network::BasePDU, [229](#)
 - ~BasePDU, [230](#)
 - IsLastFragment, [230](#)
 - Print, [230](#)
 - Read, [230](#)
 - Size, [230](#)
 - Write, [231](#)
- gdcmm::network::CEchoRQ, [289](#)

- AffectedSOPClassUID, 291
- ConstructPDV, 291
- MessageID, 291
- gdcmm::network::CEchoRSP, 291
 - ConstructPDVByDataSet, 292
- gdcmm::network::CFind, 292
- gdcmm::network::CFindCancelRQ, 293
 - ConstructPDVByDataSet, 294
- gdcmm::network::CFindRQ, 294
 - ConstructPDV, 295
- gdcmm::network::CFindRSP, 296
 - ConstructPDVByDataSet, 297
- gdcmm::network::CMoveCancelRq, 304
 - ConstructPDVByDataSet, 305
- gdcmm::network::CMoveRQ, 306
 - ConstructPDV, 307
- gdcmm::network::CMoveRSP, 307
 - ConstructPDVByDataSet, 308
- gdcmm::network::CompositeMessageFactory, 324
 - ConstructCEchoRQ, 325
 - ConstructCFindRQ, 325
 - ConstructCMoveRQ, 325
 - ConstructCStoreRQ, 325
 - ConstructCStoreRSP, 325
- gdcmm::network::CStoreRQ, 361
 - ConstructPDV, 362
- gdcmm::network::CStoreRSP, 363
 - ConstructPDV, 364
- gdcmm::network::DIMSE, 439
 - C_CANCEL_RQ, 440
 - C_ECHO_RQ, 440
 - C_ECHO_RSP, 440
 - C_FIND_RQ, 440
 - C_FIND_RSP, 440
 - C_GET_RQ, 440
 - C_GET_RSP, 440
 - C_MOVE_RQ, 440
 - C_MOVE_RSP, 440
 - C_STORE_RQ, 440
 - C_STORE_RSP, 440
 - CommandTypes, 440
 - N_ACTION_RQ, 440
 - N_ACTION_RSP, 440
 - N_CREATE_RQ, 440
 - N_CREATE_RSP, 440
 - N_DELETE_RQ, 440
 - N_DELETE_RSP, 440
 - N_EVENT_REPORT_RQ, 440
 - N_EVENT_REPORT_RSP, 440
 - N_GET_RQ, 440
 - N_GET_RSP, 440
 - N_SET_RQ, 440
 - N_SET_RSP, 440
- gdcmm::network::ImplementationClassUIDSub, 697
 - ImplementationClassUIDSub, 698
 - Print, 698
 - Read, 698
 - Size, 698
 - Write, 698
- gdcmm::network::ImplementationUIDSub, 699
 - ImplementationUIDSub, 699
 - Write, 699
- gdcmm::network::ImplementationVersionNameSub, 699
 - ImplementationVersionNameSub, 700
 - Print, 700
 - Read, 700
 - Size, 700
 - Write, 700
- gdcmm::network::MaximumLengthSub, 791
 - GetMaximumLength, 792
 - MaximumLengthSub, 792
 - Print, 792
 - Read, 792
 - SetMaximumLength, 792
 - Size, 792
 - Write, 792
- gdcmm::network::NActionRQ, 846
 - ConstructPDV, 847
- gdcmm::network::NActionRSP, 847
 - ConstructPDVByDataSet, 848
- gdcmm::network::NCreateRQ, 849
 - ConstructPDV, 850
- gdcmm::network::NCreateRSP, 850
 - ConstructPDVByDataSet, 851
- gdcmm::network::NDeleteRQ, 852
 - ConstructPDV, 853
- gdcmm::network::NDeleteRSP, 853
 - ConstructPDVByDataSet, 854
- gdcmm::network::NEventReportRQ, 858
 - ConstructPDV, 859
- gdcmm::network::NEventReportRSP, 859
 - ConstructPDVByDataSet, 860
- gdcmm::network::NGetRQ, 861
 - ConstructPDV, 862
- gdcmm::network::NGetRSP, 862
 - ConstructPDVByDataSet, 863
- gdcmm::network::NormalizedMessageFactory, 865
 - ConstructNAction, 865
 - ConstructNCreate, 865
 - ConstructNDelete, 865
 - ConstructNEventReport, 866
 - ConstructNGet, 866
 - ConstructNSet, 866
- gdcmm::network::NSetRQ, 869
 - ConstructPDV, 870
- gdcmm::network::NSetRSP, 870
 - ConstructPDVByDataSet, 871

- gdcmm::network::PDataTFPDU, 905
 - AddPresentationDataValue, 906
 - GetNumberOfPresentationDataValues, 906
 - GetPresentationDataValue, 906
 - IsLastFragment, 907
 - PDataTFPDU, 906
 - Print, 907
 - Read, 907
 - ReadInto, 907
 - Size, 907
 - SizeType, 906
 - Write, 907
- gdcmm::network::PDUFactory, 916
 - ConstructAbortPDU, 917
 - ConstructPDU, 917
 - ConstructReleasePDU, 917
 - CreateCEchoPDU, 918
 - CreateCFindPDU, 918
 - CreateCMovePDU, 918
 - CreateCStoreRQPDU, 918
 - CreateCStoreRSPPDU, 918
 - CreateNActionPDU, 918
 - CreateNCreatePDU, 918
 - CreateNDeletePDU, 919
 - CreateNEventReportPDU, 919
 - CreateNGetPDU, 919
 - CreateNSetPDU, 919
 - DetermineEventByPDU, 919
 - GetPDVs, 919
- gdcmm::network::PresentationContextAC, 970
 - GetPresentationContextID, 971
 - GetReason, 971
 - GetTransferSyntax, 971
 - PresentationContextAC, 971
 - Print, 971
 - Read, 972
 - SetPresentationContextID, 972
 - SetReason, 972
 - SetTransferSyntax, 972
 - Size, 972
 - Write, 972
- gdcmm::network::PresentationContextRQ, 975
 - AddTransferSyntax, 977
 - GetAbstractSyntax, 977
 - GetNumberOfTransferSyntaxes, 977
 - GetPresentationContextID, 977
 - GetTransferSyntax, 977
 - GetTransferSyntaxes, 978
 - operator==, 978
 - PresentationContextRQ, 976, 977
 - Print, 978
 - Read, 978
 - SetAbstractSyntax, 978
 - SetPresentationContextID, 978
 - Size, 978
 - SizeType, 976
 - Write, 978
- gdcmm::network::PresentationDataValue, 979
 - ConcatenatePDVBlobs, 980
 - ConcatenatePDVBlobsAsExplicit, 980
 - GetBlob, 980
 - GetIsCommand, 980
 - GetIsLastFragment, 980
 - GetMessageHeader, 980
 - GetPresentationContextID, 980
 - PresentationDataValue, 980
 - Print, 981
 - Read, 981
 - ReadInto, 981
 - SetBlob, 981
 - SetCommand, 981
 - SetDataSet, 981
 - SetLastFragment, 981
 - SetMessageHeader, 982
 - SetPresentationContextID, 982
 - Size, 982
 - Write, 982
- gdcmm::network::RoleSelectionSub, 1047
 - Print, 1048
 - Read, 1048
 - RoleSelectionSub, 1047
 - SetTuple, 1048
 - Size, 1048
 - Write, 1048
- gdcmm::network::ServiceClassApplicationInformation, 1114
 - Print, 1115
 - Read, 1115
 - ServiceClassApplicationInformation, 1114
 - SetTuple, 1115
 - Size, 1115
 - Write, 1115
- gdcmm::network::SOPClassExtendedNegotiationSub, 1140
 - Print, 1141
 - Read, 1141
 - SetTuple, 1141
 - Size, 1141
 - SOPClassExtendedNegotiationSub, 1141
 - Write, 1141
- gdcmm::network::TableRow, 1244
 - ~TableRow, 1244
 - TableRow, 1244
 - transitions, 1245
- gdcmm::network::TransferSyntaxSub, 1275
 - GetName, 1276
 - operator==, 1276
 - Print, 1276

- Read, [1276](#)
- SetName, [1276](#)
- SetNameFromUID, [1276](#)
- Size, [1276](#)
- TransferSyntaxSub, [1276](#)
- Write, [1277](#)
- gdcmm::network::Transition, [1277](#)
 - ~Transition, [1278](#)
 - mAction, [1278](#)
 - MakeNew, [1278](#)
 - mEnd, [1278](#)
 - Transition, [1278](#)
- gdcmm::network::ULAction, [1329](#)
 - ~ULAction, [1331](#)
 - operator=, [1332](#)
 - PerformAction, [1332](#)
 - ULAction, [1331](#)
- gdcmm::network::ULActionAA1, [1333](#)
 - PerformAction, [1334](#)
- gdcmm::network::ULActionAA2, [1334](#)
 - PerformAction, [1335](#)
- gdcmm::network::ULActionAA3, [1335](#)
 - PerformAction, [1336](#)
- gdcmm::network::ULActionAA4, [1337](#)
 - PerformAction, [1338](#)
- gdcmm::network::ULActionAA5, [1338](#)
 - PerformAction, [1339](#)
- gdcmm::network::ULActionAA6, [1339](#)
 - PerformAction, [1340](#)
- gdcmm::network::ULActionAA7, [1341](#)
 - PerformAction, [1342](#)
- gdcmm::network::ULActionAA8, [1342](#)
 - PerformAction, [1343](#)
- gdcmm::network::ULActionAE1, [1343](#)
 - PerformAction, [1344](#)
- gdcmm::network::ULActionAE2, [1345](#)
 - PerformAction, [1346](#)
- gdcmm::network::ULActionAE3, [1346](#)
 - PerformAction, [1347](#)
- gdcmm::network::ULActionAE4, [1347](#)
 - PerformAction, [1348](#)
- gdcmm::network::ULActionAE5, [1349](#)
 - PerformAction, [1350](#)
- gdcmm::network::ULActionAE6, [1350](#)
 - PerformAction, [1351](#)
- gdcmm::network::ULActionAE7, [1351](#)
 - PerformAction, [1352](#)
- gdcmm::network::ULActionAE8, [1353](#)
 - PerformAction, [1354](#)
- gdcmm::network::ULActionAR1, [1354](#)
 - PerformAction, [1355](#)
- gdcmm::network::ULActionAR10, [1355](#)
 - PerformAction, [1356](#)
- gdcmm::network::ULActionAR2, [1357](#)
 - PerformAction, [1358](#)
- gdcmm::network::ULActionAR3, [1358](#)
 - PerformAction, [1359](#)
- gdcmm::network::ULActionAR4, [1359](#)
 - PerformAction, [1360](#)
- gdcmm::network::ULActionAR5, [1361](#)
 - PerformAction, [1362](#)
- gdcmm::network::ULActionAR6, [1362](#)
 - PerformAction, [1363](#)
- gdcmm::network::ULActionAR7, [1363](#)
 - PerformAction, [1364](#)
- gdcmm::network::ULActionAR8, [1365](#)
 - PerformAction, [1366](#)
- gdcmm::network::ULActionAR9, [1366](#)
 - PerformAction, [1367](#)
- gdcmm::network::ULActionDT1, [1367](#)
 - PerformAction, [1368](#)
- gdcmm::network::ULActionDT2, [1369](#)
 - PerformAction, [1370](#)
- gdcmm::network::ULBasicCallback, [1370](#)
 - ~ULBasicCallback, [1372](#)
 - GetDataSets, [1372](#)
 - GetResponses, [1372](#)
 - HandleDataSet, [1372](#)
 - HandleResponse, [1372](#)
 - ULBasicCallback, [1372](#)
- gdcmm::network::ULConnection, [1373](#)
 - ~ULConnection, [1374](#)
 - AddAcceptedPresentationContext, [1374](#)
 - FindContext, [1374](#)
 - GetAcceptedPresentationContexts, [1375](#)
 - GetConnectionInfo, [1375](#)
 - GetMaxPDUSize, [1375](#)
 - GetPresentationContextACByID, [1375](#)
 - GetPresentationContextIDFromPresentationContext, [1375](#)
 - GetPresentationContextRQByID, [1375](#)
 - GetPresentationContexts, [1375](#)
 - GetProtocol, [1375](#)
 - GetState, [1376](#)
 - GetTimer, [1376](#)
 - InitializeConnection, [1376](#)
 - InitializeIncomingConnection, [1376](#)
 - operator=, [1376](#)
 - SetMaxPDUSize, [1376](#)
 - SetPresentationContexts, [1376](#)
 - SetState, [1377](#)
 - StopProtocol, [1377](#)
 - ULActionAE6, [1377](#)
 - ULConnection, [1374](#)
 - ULConnectionManager, [1377](#)
- gdcmm::network::ULConnectionCallback, [1378](#)
 - ~ULConnectionCallback, [1379](#)
 - DataSetHandled, [1379](#)

- DataSetHandles, [1379](#)
- HandleDataSet, [1379](#)
- HandleResponse, [1379](#)
- mImplicit, [1380](#)
- ResetHandledDataSet, [1379](#)
- SetImplicitFlag, [1379](#)
- ULConnectionCallback, [1379](#)
- gdcmm::network::ULConnectionInfo, [1380](#)
 - GetCalledAETitle, [1381](#)
 - GetCalledComputerName, [1381](#)
 - GetCalledIPAddress, [1381](#)
 - GetCalledIPPort, [1381](#)
 - GetCallingAETitle, [1381](#)
 - GetMaxPDULength, [1381](#)
 - Initialize, [1381](#)
 - SetMaxPDULength, [1381](#)
 - ULConnectionInfo, [1381](#)
- gdcmm::network::ULConnectionManager, [1382](#)
 - ~ULConnectionManager, [1385](#)
 - BreakConnection, [1385](#)
 - BreakConnectionNow, [1385](#)
 - EstablishConnection, [1385](#)
 - EstablishConnectionMove, [1386](#)
 - mConnection, [1389](#)
 - mSecondaryConnection, [1389](#)
 - mTransitions, [1389](#)
 - RunEventLoop, [1386](#)
 - RunMoveEventLoop, [1386](#)
 - SendEcho, [1386](#)
 - SendFind, [1386](#), [1387](#)
 - SendMove, [1387](#)
 - SendNAction, [1387](#)
 - SendNCreate, [1387](#)
 - SendNDelete, [1388](#)
 - SendNEventReport, [1388](#)
 - SendNGet, [1388](#)
 - SendNSet, [1388](#)
 - SendStore, [1389](#)
 - ULConnectionManager, [1385](#)
- gdcmm::network::ULEvent, [1390](#)
 - ~ULEvent, [1390](#)
 - GetDataSetPos, [1391](#)
 - GetEvent, [1391](#)
 - GetIStream, [1391](#)
 - GetPDUs, [1391](#)
 - SetEvent, [1391](#)
 - SetPDU, [1391](#)
 - ULEvent, [1390](#)
- gdcmm::network::ULTransitionTable, [1391](#)
 - HandleEvent, [1392](#)
 - PrintTable, [1392](#)
 - ULTransitionTable, [1392](#)
- gdcmm::network::ULWritingCallback, [1393](#)
 - ~ULWritingCallback, [1394](#)
- HandleDataSet, [1394](#)
- HandleResponse, [1394](#)
- SetDirectory, [1394](#)
- ULWritingCallback, [1394](#)
- gdcmm::network::UserInfo, [1408](#)
 - ~UserInfo, [1409](#)
 - AddRoleSelectionSub, [1409](#)
 - AddSOPClassExtendedNegotiationSub, [1409](#)
 - GetMaximumLengthSub, [1409](#)
 - operator=, [1409](#)
 - Print, [1410](#)
 - Read, [1410](#)
 - Size, [1410](#)
 - UserInfo, [1409](#)
 - Write, [1410](#)
- gdcmm::NoEvent, [864](#)
- gdcmm::NormalizedNetworkFunctions, [866](#)
 - ConstructQuery, [867](#)
 - NAction, [867](#)
 - NCreate, [867](#)
 - NDelete, [868](#)
 - NEventReport, [868](#)
 - NGet, [868](#)
 - NSet, [868](#)
- gdcmm::Object, [872](#)
 - ~Object, [873](#)
 - Object, [873](#)
 - operator<<, [874](#)
 - operator=, [874](#)
 - Print, [874](#)
 - Register, [874](#)
 - SmartPointer, [874](#)
 - UnRegister, [874](#)
- gdcmm::OpenSSLCryptoFactory, [875](#)
 - CreateCMSProvider, [877](#)
 - InitOpenSSL, [877](#)
 - OpenSSLCryptoFactory, [876](#)
- gdcmm::OpenSSLCryptographicMessageSyntax, [877](#)
 - ~OpenSSLCryptographicMessageSyntax, [879](#)
 - Decrypt, [879](#)
 - Encrypt, [879](#)
 - GetCipherType, [879](#)
 - OpenSSLCryptographicMessageSyntax, [879](#)
 - ParseCertificateFile, [879](#)
 - ParseKeyFile, [880](#)
 - SetCipherType, [880](#)
 - SetPassword, [880](#)
- gdcmm::OpenSSLP7CryptoFactory, [881](#)
 - CreateCMSProvider, [882](#)
 - OpenSSLP7CryptoFactory, [882](#)
- gdcmm::OpenSSLP7CryptographicMessageSyntax, [883](#)
 - ~OpenSSLP7CryptographicMessageSyntax, [884](#)
 - Decrypt, [884](#)

- Encrypt, 884
- GetCipherType, 885
- OpenSSLP7CryptographicMessageSyntax, 884
- ParseCertificateFile, 885
- ParseKeyFile, 885
- SetCipherType, 885
- SetPassword, 885
- gdcm::Orientation, 886
 - ~Orientation, 887
 - AXIAL, 887
 - CORONAL, 887
 - GetLabel, 888
 - GetMajorAxisFromPatientRelativeDirectionCosine, 888
 - GetObliquityThresholdCosineValue, 888
 - GetType, 888
 - OBLIQUE, 887
 - operator<<, 889
 - Orientation, 887
 - OrientationType, 887
 - Print, 888
 - SAGITTAL, 887
 - SetObliquityThresholdCosineValue, 888
 - UNKNOWN, 887
- gdcm::Overlay, 889
 - ~Overlay, 892
 - Decompress, 893
 - GetBitPosition, 893
 - GetBitsAllocated, 893
 - GetColumns, 893
 - GetDescription, 893
 - GetGroup, 893
 - GetOrigin, 894
 - GetOverlayData, 894
 - GetOverlayTypeAsString, 894
 - GetOverlayTypeFromString, 894
 - GetRows, 894
 - GetType, 894
 - GetTypeAsEnum, 894
 - GetUnpackBuffer, 894
 - GetUnpackBufferLength, 895
 - GrabOverlayFromPixelData, 895
 - Graphics, 892
 - Invalid, 892
 - IsEmpty, 895
 - IsInPixelData, 895
 - IsZero, 895
 - operator=, 896
 - Overlay, 892, 893
 - OverlayType, 892
 - Print, 896
 - ROI, 892
 - SetBitPosition, 896
 - SetBitsAllocated, 896
 - SetColumns, 896
 - SetDescription, 896
 - SetFrameOrigin, 897
 - SetGroup, 897
 - SetNumberOfFrames, 897
 - SetOrigin, 897
 - SetOverlay, 897
 - SetRows, 897
 - SetType, 898
 - Update, 898
- gdcm::ParseException, 898
 - ~ParseException, 899
 - GetLastElement, 900
 - operator=, 900
 - ParseException, 899, 900
 - SetLastElement, 900
- gdcm::Parser, 900
 - ~Parser, 902
 - DuplicateAttributeError, 902
 - EndElementHandler, 902
 - ErrorType, 902
 - GetBuffer, 903
 - GetCurrentByteIndex, 903
 - GetErrorCode, 903
 - GetErrorString, 903
 - GetUserData, 903
 - JunkAfterDocElementError, 902
 - NoElementsError, 902
 - NoError, 902
 - NoMemoryError, 902
 - Parse, 903
 - ParseBuffer, 903
 - Parser, 902
 - Process, 903
 - SetElementHandler, 903
 - SetUserData, 904
 - StartElementHandler, 902
 - SyntaxError, 902
 - TagMismatchError, 902
 - UndefinedEntityError, 902
 - UnexpectedStateError, 902
- gdcm::Patient, 904
 - Patient, 904
- gdcm::PDBelement, 908
 - GetName, 909
 - GetValue, 909
 - NameField, 910
 - operator<<, 910
 - operator==, 909
 - PDBelement, 909
 - SetName, 910
 - SetValue, 910
 - ValueField, 910
- gdcm::PDBHeader, 911

- ~PDBHeader, [912](#)
- FindPDBelementByName, [912](#)
- GetPDBeEnd, [912](#)
- GetPDBelementByName, [912](#)
- GetPDBInfoTag, [913](#)
- LoadFromDataElement, [913](#)
- operator<<, [913](#)
- PDBHeader, [912](#)
- Print, [913](#)
- gdcmm::PDFCodec, [914](#)
 - ~PDFCodec, [915](#)
 - CanCode, [916](#)
 - CanDecode, [916](#)
 - Decode, [916](#)
 - PDFCodec, [915](#)
- gdcmm::PersonName, [920](#)
 - Component, [921](#)
 - GetMaxLength, [920](#)
 - GetNumberOfComponents, [920](#)
 - MaxLength, [921](#)
 - MaxNumberOfComponents, [921](#)
 - Padding, [922](#)
 - Print, [920](#)
 - Separator, [922](#)
 - SetBlob, [921](#)
 - SetComponents, [921](#)
- gdcmm::PGXCodec, [922](#)
 - ~PGXCodec, [925](#)
 - CanCode, [925](#)
 - CanDecode, [925](#)
 - Clone, [925](#)
 - GetHeaderInfo, [925](#)
 - PGXCodec, [925](#)
 - Read, [926](#)
 - Write, [926](#)
- gdcmm::PhotometricInterpretation, [926](#)
 - ARGB, [928](#)
 - CMYK, [928](#)
 - GetPIString, [928](#)
 - GetPIType, [928](#)
 - GetSamplesPerPixel, [928](#)
 - GetString, [929](#)
 - GetType, [929](#)
 - HSV, [928](#)
 - IsLossless, [929](#)
 - IsLossy, [929](#)
 - IsRetired, [929](#)
 - IsSameColorSpace, [929](#)
 - MONOCHROME1, [927](#)
 - MONOCHROME2, [927](#)
 - operator PIType, [929](#)
 - operator<<, [930](#)
 - PALETTE_COLOR, [928](#)
 - PhotometricInterpretation, [928](#)
 - PI_END, [928](#)
 - PIType, [927](#)
 - RGB, [928](#)
 - UNKNOWN, [927](#)
 - YBR_FULL, [928](#)
 - YBR_FULL_422, [928](#)
 - YBR_ICT, [928](#)
 - YBR_PARTIAL_420, [928](#)
 - YBR_PARTIAL_422, [928](#)
 - YBR_RCT, [928](#)
- gdcmm::PixelFormat, [930](#)
 - Bitmap, [938](#)
 - FLOAT16, [933](#)
 - FLOAT32, [933](#)
 - FLOAT64, [933](#)
 - GetBitsAllocated, [933](#)
 - GetBitsStored, [933](#)
 - GetHighBit, [934](#)
 - GetMax, [934](#)
 - GetMin, [934](#)
 - GetPixelRepresentation, [934](#)
 - GetPixelSize, [934](#)
 - GetSamplesPerPixel, [935](#)
 - GetScalarType, [935](#)
 - GetScalarTypeAsString, [935](#)
 - INT12, [932](#)
 - INT16, [932](#)
 - INT32, [932](#)
 - INT64, [933](#)
 - INT8, [932](#)
 - IsCompatible, [935](#)
 - IsValid, [936](#)
 - operator ScalarType, [936](#)
 - operator!=, [936](#)
 - operator<<, [938](#)
 - operator==, [936](#)
 - PixelFormat, [933](#)
 - Print, [936](#)
 - ScalarType, [932](#)
 - SetBitsAllocated, [937](#)
 - SetBitsStored, [937](#)
 - SetHighBit, [937](#)
 - SetPixelRepresentation, [937](#)
 - SetSamplesPerPixel, [937](#)
 - SetScalarType, [937](#)
 - SINGLEBIT, [933](#)
 - UINT12, [932](#)
 - UINT16, [932](#)
 - UINT32, [932](#)
 - UINT64, [932](#)
 - UINT8, [932](#)
 - UNKNOWN, [933](#)
 - Validate, [938](#)
- gdcmm::Pixmap, [939](#)

- [~Pixmap, 943](#)
 - [AreOverlaysInPixelData, 943](#)
 - [Curves, 945](#)
 - [GetCurve, 943](#)
 - [GetIconImage, 943](#)
 - [GetNumberOfCurves, 944](#)
 - [GetNumberOfOverlays, 944](#)
 - [GetOverlay, 944](#)
 - [Icon, 945](#)
 - [Overlays, 946](#)
 - [Pixmap, 943](#)
 - [Print, 944](#)
 - [RemoveOverlay, 944](#)
 - [SetIconImage, 945](#)
 - [SetNumberOfCurves, 945](#)
 - [SetNumberOfOverlays, 945](#)
 - [UnusedBitsPresentInPixelData, 945](#)
- [gdcm::PixmapReader, 946](#)
 - [~PixmapReader, 949](#)
 - [GetPixmap, 949](#)
 - [PixelData, 950](#)
 - [PixmapReader, 949](#)
 - [Read, 949](#)
 - [ReadACRNEMAIImage, 949](#)
 - [ReadImage, 950](#)
 - [ReadImageInternal, 950](#)
- [gdcm::PixmapToPixmapFilter, 950](#)
 - [~PixmapToPixmapFilter, 952](#)
 - [GetInput, 952](#)
 - [GetOutput, 952](#)
 - [GetOutputAsPixmap, 952](#)
 - [PixmapToPixmapFilter, 952](#)
- [gdcm::PixmapWriter, 953](#)
 - [~PixmapWriter, 956](#)
 - [DoIconImage, 956](#)
 - [GetImage, 956](#)
 - [GetPixmap, 956](#)
 - [PixelData, 958](#)
 - [PixmapWriter, 956](#)
 - [PrepareWrite, 957](#)
 - [SetImage, 957](#)
 - [SetPixmap, 957](#)
 - [Write, 957](#)
- [gdcm::PNMCodec, 958](#)
 - [~PNMCodec, 961](#)
 - [CanCode, 961](#)
 - [CanDecode, 961](#)
 - [Clone, 962](#)
 - [GetBufferLength, 962](#)
 - [GetHeaderInfo, 962](#)
 - [PNMCodec, 961](#)
 - [Read, 962](#)
 - [SetBufferLength, 962](#)
 - [Write, 962](#)
- [gdcm::Preamble, 963](#)
 - [~Preamble, 964](#)
 - [Clear, 964](#)
 - [Create, 964](#)
 - [GetInternal, 965](#)
 - [GetLength, 965](#)
 - [IsEmpty, 965](#)
 - [IsValid, 965](#)
 - [operator<<, 966](#)
 - [operator=, 965](#)
 - [Preamble, 964](#)
 - [Print, 965](#)
 - [Read, 965](#)
 - [Remove, 966](#)
 - [Valid, 966](#)
 - [Write, 966](#)
- [gdcm::PresentationContext, 967](#)
 - [AbstractSyntax, 970](#)
 - [AddTransferSyntax, 969](#)
 - [GetAbstractSyntax, 969](#)
 - [GetNumberOfTransferSyntaxes, 969](#)
 - [GetPresentationContextID, 969](#)
 - [GetTransferSyntax, 969](#)
 - [ID, 970](#)
 - [operator==, 969](#)
 - [PresentationContext, 968](#)
 - [Print, 969](#)
 - [SetAbstractSyntax, 969](#)
 - [SetPresentationContextID, 970](#)
 - [SizeType, 968](#)
 - [TransferSyntaxArrayType, 968](#)
 - [TransferSyntaxes, 970](#)
- [gdcm::PresentationContextGenerator, 972](#)
 - [AddFromFile, 974](#)
 - [AddPresentationContext, 974](#)
 - [GenerateFromFilenames, 974](#)
 - [GenerateFromUID, 974](#)
 - [GetDefaultTransferSyntax, 975](#)
 - [GetPresentationContexts, 975](#)
 - [PresentationContextArrayType, 974](#)
 - [PresentationContextGenerator, 974](#)
 - [SetDefaultTransferSyntax, 975](#)
 - [SetMergeModeToAbstractSyntax, 975](#)
 - [SetMergeModeToTransferSyntax, 975](#)
 - [SizeType, 974](#)
- [gdcm::Printer, 983](#)
 - [~Printer, 985](#)
 - [CONDENSED_STYLE, 985](#)
 - [CXX, 985](#)
 - [F, 987](#)
 - [GetPrintStyle, 985](#)
 - [MaxPrintLength, 987](#)
 - [Print, 985](#)
 - [PrintDataElement, 985](#)

- PrintDataSet, 986
- Printer, 985
- PrintSQ, 986
- PrintStyle, 987
- PrintStyles, 985
- SetColor, 986
- SetFile, 986
- SetStyle, 986
- VERBOSE_STYLE, 985
- XML, 985
- gdcmm::PrivateDict, 987
 - ~PrivateDict, 988
 - AddDictEntry, 988
 - Dicts, 989
 - FindDictEntry, 988
 - GetDictEntry, 988
 - IsEmpty, 989
 - LoadDefault, 989
 - operator<<, 989
 - PrintXML, 989
 - PrivateDict, 988
 - RemoveDictEntry, 989
- gdcmm::PrivateTag, 990
 - GetAsDataElement, 993
 - GetOwner, 993
 - operator!=, 993
 - operator<, 994
 - operator<<, 995
 - operator=, 994
 - operator==, 994
 - PrivateTag, 993
 - ReadFromCommaSeparatedString, 994
 - SetOwner, 994
- gdcmm::ProgressEvent, 995
 - ~ProgressEvent, 997
 - CheckEvent, 998
 - GetEventName, 998
 - GetProgress, 998
 - MakeObject, 998
 - operator=, 998
 - ProgressEvent, 997
 - Self, 997
 - SetProgress, 998
 - Superclass, 997
- gdcmm::PVRGCodec, 999
 - ~PVRGCodec, 1002
 - CanCode, 1002
 - CanDecode, 1002
 - Clone, 1002
 - Code, 1003
 - Decode, 1003
 - PVRGCodec, 1002
 - SetLossyFlag, 1003
- gdcmm::PythonFilter, 1003
 - ~PythonFilter, 1004
 - GetFile, 1004
 - PythonFilter, 1004
 - SetDicts, 1004
 - SetFile, 1004
 - ToPyObject, 1005
 - UseDictAlways, 1005
- gdcmm::QueryBase, 1005
 - ~QueryBase, 1006
 - GetAllRequiredTags, 1006
 - GetAllTags, 1006
 - GetHierarchicalSearchTags, 1006
 - GetName, 1007
 - GetOptionalTags, 1007
 - GetQueryLevel, 1007
 - GetRequiredTags, 1007
 - GetUniqueTags, 1007
- gdcmm::QueryFactory, 1008
 - GetCharacterFromCurrentLocale, 1008
 - ListCharSets, 1008
 - ProduceCharacterSetDataElement, 1008
 - ProduceQuery, 1009
- gdcmm::QueryImage, 1009
 - GetHierarchicalSearchTags, 1010
 - GetName, 1010
 - GetOptionalTags, 1011
 - GetQueryLevel, 1011
 - GetRequiredTags, 1011
 - GetUniqueTags, 1011
- gdcmm::QueryPatient, 1012
 - GetHierarchicalSearchTags, 1013
 - GetName, 1013
 - GetOptionalTags, 1013
 - GetQueryLevel, 1013
 - GetRequiredTags, 1013
 - GetUniqueTags, 1014
- gdcmm::QuerySeries, 1014
 - GetHierarchicalSearchTags, 1015
 - GetName, 1015
 - GetOptionalTags, 1016
 - GetQueryLevel, 1016
 - GetRequiredTags, 1016
 - GetUniqueTags, 1016
- gdcmm::QueryStudy, 1017
 - GetHierarchicalSearchTags, 1018
 - GetName, 1018
 - GetOptionalTags, 1018
 - GetQueryLevel, 1018
 - GetRequiredTags, 1018
 - GetUniqueTags, 1019
- gdcmm::RAWCodec, 1019
 - ~RAWCodec, 1022
 - CanCode, 1022
 - CanDecode, 1022

- Clone, [1022](#)
- Code, [1023](#)
- Decode, [1023](#)
- DecodeByStreams, [1023](#)
- DecodeBytes, [1023](#)
- GetHeaderInfo, [1023](#)
- RAWCodec, [1022](#)
- gdcm::Reader, [1024](#)
 - ~Reader, [1027](#)
 - CanRead, [1027](#)
 - F, [1031](#)
 - GetFile, [1027](#)
 - GetStreamCurrentPosition, [1028](#)
 - GetStreamPtr, [1028](#)
 - Read, [1028](#)
 - ReadDataSet, [1029](#)
 - Reader, [1027](#)
 - ReadMetaInformation, [1029](#)
 - ReadPreamble, [1029](#)
 - ReadSelectedPrivateTags, [1029](#)
 - ReadSelectedTags, [1029](#)
 - ReadUpToTag, [1030](#)
 - SetFile, [1030](#)
 - SetFileName, [1030](#)
 - SetStream, [1031](#)
 - StreamImageReader, [1031](#)
- gdcm::RealWorldValueMappingContent, [1032](#)
 - CodeMeaning, [1032](#)
 - CodeValue, [1032](#)
 - RealWorldValueIntercept, [1032](#)
 - RealWorldValueSlope, [1033](#)
- gdcm::Region, [1033](#)
 - ~Region, [1034](#)
 - Area, [1034](#)
 - Clone, [1034](#)
 - ComputeBoundingBox, [1034](#)
 - Empty, [1034](#)
 - IsValid, [1035](#)
 - Print, [1035](#)
 - Region, [1034](#)
- gdcm::Rescaler, [1035](#)
 - ~Rescaler, [1037](#)
 - ComputeInterceptSlopePixelType, [1037](#)
 - ComputePixelTypeFromMinMax, [1037](#)
 - GetIntercept, [1038](#)
 - GetSlope, [1038](#)
 - InverseRescale, [1038](#)
 - InverseRescaleFunctionIntoBestFit, [1038](#)
 - Rescale, [1038](#)
 - RescaleFunctionIntoBestFit, [1038](#)
 - Rescaler, [1037](#)
 - SetIntercept, [1039](#)
 - SetMinMaxForPixelType, [1039](#)
 - SetPixelFormat, [1039](#)
 - SetSlope, [1039](#)
 - SetTargetPixelType, [1039](#)
 - SetUseTargetPixelType, [1040](#)
- gdcm::RLECodec, [1040](#)
 - ~RLECodec, [1043](#)
 - AppendFrameEncode, [1044](#)
 - AppendRowEncode, [1044](#)
 - CanCode, [1044](#)
 - CanDecode, [1044](#)
 - Clone, [1044](#)
 - Code, [1044](#)
 - Decode, [1045](#)
 - DecodeByStreams, [1045](#)
 - DecodeExtent, [1045](#)
 - GetBufferLength, [1045](#)
 - GetHeaderInfo, [1045](#)
 - ImageRegionReader, [1047](#)
 - IsFrameEncoder, [1046](#)
 - IsRowEncoder, [1046](#)
 - RLECodec, [1043](#)
 - SetBufferLength, [1046](#)
 - SetLength, [1046](#)
 - StartEncode, [1046](#)
 - StopEncode, [1046](#)
- gdcm::Scanner, [1049](#)
 - ~Scanner, [1053](#)
 - AddPrivateTag, [1053](#)
 - AddSkipTag, [1053](#)
 - AddTag, [1053](#)
 - Begin, [1054](#)
 - ClearSkipTags, [1054](#)
 - ClearTags, [1054](#)
 - ConstIterator, [1052](#)
 - End, [1054](#)
 - GetAllFileNamesFromTagToValue, [1054](#)
 - GetFilenameFromTagToValue, [1054](#)
 - GetFileNames, [1054](#)
 - GetKeys, [1055](#)
 - GetMapping, [1055](#)
 - GetMappingFromTagToValue, [1055](#)
 - GetMappings, [1055](#)
 - GetOrderedValues, [1055](#)
 - GetValue, [1055](#)
 - GetValues, [1056](#)
 - IsKey, [1056](#)
 - MappingType, [1052](#)
 - New, [1056](#)
 - operator<<, [1058](#)
 - Print, [1057](#)
 - PrintTable, [1057](#)
 - ProcessPublicTag, [1057](#)
 - Scan, [1057](#)
 - Scanner, [1053](#)
 - TagToValue, [1052](#)

- TagToValueValueType, [1052](#)
- ValueType, [1053](#)
- gdcm::Scanner2, [1058](#)
 - ~Scanner2, [1063](#)
 - AddPrivateTag, [1063](#)
 - AddPublicTag, [1063](#)
 - AddSkipTag, [1063](#)
 - Begin, [1064](#)
 - ClearPrivateTags, [1064](#)
 - ClearPublicTags, [1064](#)
 - ClearSkipTags, [1064](#)
 - End, [1064](#)
 - GetAllFilenamesFromPrivateTagToValue, [1064](#)
 - GetAllFilenamesFromPublicTagToValue, [1064](#)
 - GetFilenameFromPrivateTagToValue, [1064](#)
 - GetFilenameFromPublicTagToValue, [1065](#)
 - GetFilenames, [1065](#)
 - GetKeys, [1065](#)
 - GetMappingFromPrivateTagToValue, [1065](#)
 - GetMappingFromPublicTagToValue, [1065](#)
 - GetPrivateMapping, [1065](#)
 - GetPrivateMappings, [1065](#)
 - GetPrivateOrderedValues, [1066](#)
 - GetPrivateValue, [1066](#)
 - GetPrivateValues, [1066](#)
 - GetPublicMapping, [1066](#)
 - GetPublicMappings, [1066](#)
 - GetPublicOrderedValues, [1066](#)
 - GetPublicValue, [1066](#)
 - GetPublicValues, [1067](#)
 - GetValues, [1067](#)
 - IsKey, [1067](#)
 - New, [1067](#)
 - operator<<, [1069](#)
 - Print, [1067](#)
 - PrintTable, [1068](#)
 - PrivateBegin, [1068](#)
 - PrivateConstIterator, [1062](#)
 - PrivateEnd, [1068](#)
 - PrivateMappingType, [1062](#)
 - PrivateTagToValue, [1062](#)
 - PrivateTagToValueValueType, [1062](#)
 - ProcessPrivateTag, [1068](#)
 - ProcessPublicTag, [1068](#)
 - PublicConstIterator, [1062](#)
 - PublicMappingType, [1062](#)
 - PublicTagToValue, [1062](#)
 - PublicTagToValueValueType, [1063](#)
 - Scan, [1068](#)
 - Scanner2, [1063](#)
 - ValueType, [1063](#)
- gdcm::Scanner2::ltstr, [784](#)
 - operator(), [785](#)
- gdcm::Scanner::ltstr, [785](#)
 - operator(), [785](#)
- gdcm::Segment, [1069](#)
 - ~Segment, [1072](#)
 - AddSurface, [1072](#)
 - ALGOType, [1072](#)
 - ALGOType_END, [1072](#)
 - AnatomicRegion, [1076](#)
 - AnatomicRegionModifiers, [1076](#)
 - AUTOMATIC, [1072](#)
 - BasicCodedEntryVector, [1072](#)
 - GetALGOType, [1072](#)
 - GetALGOTypeString, [1073](#)
 - GetAnatomicRegion, [1073](#)
 - GetAnatomicRegionModifiers, [1073](#)
 - GetPropertyCategory, [1073](#)
 - GetPropertyType, [1073](#)
 - GetPropertyTypeModifiers, [1074](#)
 - GetSegmentAlgorithmName, [1074](#)
 - GetSegmentAlgorithmType, [1074](#)
 - GetSegmentDescription, [1074](#)
 - GetSegmentLabel, [1074](#)
 - GetSegmentNumber, [1074](#)
 - GetSurface, [1074](#)
 - GetSurfaceCount, [1074](#)
 - GetSurfaces, [1075](#)
 - MANUAL, [1072](#)
 - PropertyCategory, [1076](#)
 - PropertyType, [1077](#)
 - PropertyTypeModifiers, [1077](#)
 - Segment, [1072](#)
 - SegmentAlgorithmName, [1077](#)
 - SegmentAlgorithmType, [1077](#)
 - SegmentDescription, [1077](#)
 - SegmentLabel, [1077](#)
 - SegmentNumber, [1077](#)
 - SEMIAUTOMATIC, [1072](#)
 - SetAnatomicRegion, [1075](#)
 - SetAnatomicRegionModifiers, [1075](#)
 - SetPropertyCategory, [1075](#)
 - SetPropertyType, [1075](#)
 - SetPropertyTypeModifiers, [1075](#)
 - SetSegmentAlgorithmName, [1075](#)
 - SetSegmentAlgorithmType, [1075](#), [1076](#)
 - SetSegmentDescription, [1076](#)
 - SetSegmentLabel, [1076](#)
 - SetSegmentNumber, [1076](#)
 - SetSurfaceCount, [1076](#)
 - SurfaceCount, [1077](#)
 - Surfaces, [1077](#)
 - SurfaceVector, [1072](#)
- gdcm::SegmentedPaletteColorLookupTable, [1078](#)
 - ~SegmentedPaletteColorLookupTable, [1081](#)
 - Print, [1081](#)
 - SegmentedPaletteColorLookupTable, [1081](#)

- SetLUT, 1081
- gdcm::SegmentHelper, 108
- gdcm::SegmentHelper::BasicCodedEntry, 241
 - BasicCodedEntry, 242
 - CM, 243
 - CSD, 243
 - CSV, 243
 - CV, 244
 - IsEmpty, 243
- gdcm::SegmentReader, 1082
 - ~SegmentReader, 1085
 - GetSegments, 1085
 - Read, 1085
 - ReadSegment, 1086
 - ReadSegments, 1086
 - SegmentMap, 1085
 - SegmentReader, 1085
 - Segments, 1086
 - SegmentVector, 1085
- gdcm::SegmentWriter, 1086
 - ~SegmentWriter, 1090
 - AddSegment, 1090
 - GetNumberOfSegments, 1090
 - GetSegment, 1090
 - GetSegments, 1090
 - PrepareWrite, 1090
 - Segments, 1091
 - SegmentVector, 1090
 - SegmentWriter, 1090
 - SetNumberOfSegments, 1091
 - SetSegments, 1091
 - Write, 1091
- gdcm::SequenceOfFragments, 1092
 - AddFragment, 1095
 - Begin, 1095
 - Clear, 1095
 - ComputeByteLength, 1095
 - ComputeLength, 1096
 - ConstIterator, 1094
 - End, 1096
 - FragmentVector, 1094
 - GetBuffer, 1096
 - GetFragBuffer, 1096
 - GetFragment, 1096
 - GetLength, 1096
 - GetNumberOfFragments, 1097
 - GetTable, 1097
 - Iterator, 1094
 - New, 1097
 - operator==, 1097
 - Print, 1097
 - Read, 1098
 - ReadPreValue, 1098
 - ReadValue, 1098
 - SequenceOfFragments, 1095
 - SetLength, 1098
 - SizeType, 1094
 - Write, 1098
 - WriteBuffer, 1099
- gdcm::SequenceOfItems, 1099
 - AddItem, 1103
 - AddNewUndefinedLengthItem, 1103
 - Begin, 1103, 1104
 - Clear, 1104
 - ComputeLength, 1104
 - ConstIterator, 1102
 - End, 1104
 - FindDataElement, 1104
 - GetItem, 1104, 1105
 - GetLength, 1105
 - GetNumberOfItems, 1105
 - IsEmpty, 1105
 - IsUndefinedLength, 1105
 - Items, 1108
 - ItemVector, 1102
 - Iterator, 1102
 - New, 1106
 - operator=, 1106
 - operator==, 1106
 - Print, 1106
 - Read, 1106
 - RemoveItemByIndex, 1107
 - SequenceLengthField, 1108
 - SequenceOfItems, 1103
 - SetLength, 1107
 - SetLengthToUndefined, 1107
 - SetNumberOfItems, 1107
 - SizeType, 1103
 - Write, 1107
- gdcm::SerieHelper, 1108
 - ~SerieHelper, 1111
 - AddFile, 1111
 - AddFileName, 1111
 - AddRestriction, 1111
 - Clear, 1111
 - CreateDefaultUniqueSeriesIdentifier, 1112
 - CreateUniqueSeriesIdentifier, 1112
 - FileNameOrdering, 1112
 - GetFirstSingleSerieUIDFileSet, 1112
 - GetNextSingleSerieUIDFileSet, 1112
 - ImageNumberOrdering, 1112
 - ImagePositionPatientOrdering, 1112
 - ItFileSetHt, 1113
 - OrderFileList, 1112
 - Rule, 1110
 - SerieHelper, 1111
 - SerieRestrictions, 1110
 - SetDirectory, 1112

- SetLoadMode, [1113](#)
- SetUseSeriesDetails, [1113](#)
- SingleSerieUIDFileSetHT, [1113](#)
- SingleSerieUIDFileSetmap, [1110](#)
- UserOrdering, [1113](#)
- gdcm::Series, [1113](#)
 - Series, [1114](#)
- gdcm::ServiceClassUser, [1116](#)
 - ~ServiceClassUser, [1119](#)
 - GetAETitle, [1119](#)
 - GetCalledAETitle, [1119](#)
 - GetTimeout, [1119](#)
 - InitializeConnection, [1119](#)
 - IsPresentationContextAccepted, [1120](#)
 - New, [1120](#)
 - operator=, [1120](#)
 - SendEcho, [1120](#)
 - SendFind, [1120](#)
 - SendMove, [1120](#), [1121](#)
 - SendStore, [1121](#)
 - ServiceClassUser, [1119](#)
 - SetAETitle, [1122](#)
 - SetCalledAETitle, [1122](#)
 - SetHostname, [1122](#)
 - SetPort, [1122](#)
 - SetPortSCP, [1122](#)
 - SetPresentationContexts, [1123](#)
 - SetTimeout, [1123](#)
 - StartAssociation, [1123](#)
 - StopAssociation, [1123](#)
- gdcm::SHA1, [1124](#)
 - ~SHA1, [1125](#)
 - Compute, [1125](#)
 - ComputeFile, [1125](#)
 - operator=, [1125](#)
 - SHA1, [1125](#)
- gdcm::SimpleMemberCommand< T >, [1126](#)
 - ~SimpleMemberCommand, [1129](#)
 - Execute, [1130](#)
 - m_MemberFunction, [1131](#)
 - m_This, [1131](#)
 - New, [1130](#)
 - operator=, [1130](#)
 - Self, [1129](#)
 - SetCallbackFunction, [1130](#)
 - SimpleMemberCommand, [1129](#)
 - TMemberFunctionPointer, [1129](#)
- gdcm::SimpleSubjectWatcher, [1131](#)
 - ~SimpleSubjectWatcher, [1132](#)
 - EndFilter, [1132](#)
 - operator=, [1132](#)
 - ShowAbort, [1132](#)
 - ShowAnonymization, [1133](#)
 - ShowData, [1133](#)
 - ShowDataSet, [1133](#)
 - ShowFileName, [1133](#)
 - ShowIteration, [1133](#)
 - ShowProgress, [1133](#)
 - SimpleSubjectWatcher, [1132](#)
 - StartFilter, [1133](#)
 - TestAbortOff, [1134](#)
 - TestAbortOn, [1134](#)
- gdcm::SmartPointer< ObjectType >, [1136](#)
 - ~SmartPointer, [1139](#)
 - GetPointer, [1139](#)
 - operator ObjectType *, [1139](#)
 - operator->, [1139](#)
 - operator=, [1139](#), [1140](#)
 - operator*, [1139](#)
 - SmartPointer, [1138](#), [1139](#)
- gdcm::SOPClassUIDToIOD, [1142](#)
 - const, [1142](#)
 - GetIOD, [1142](#)
 - GetIODFromSOPClassUID, [1142](#)
 - GetNumberOfSOPClassToIOD, [1143](#)
 - GetSOPClassUIDFromIOD, [1143](#)
 - GetSOPClassUIDToIOD, [1143](#)
 - GetSOPClassUIDToIODs, [1143](#)
- gdcm::Sorter, [1143](#)
 - ~Sorter, [1145](#)
 - AddSelect, [1146](#)
 - Filenames, [1147](#)
 - GetFilenames, [1146](#)
 - operator<<, [1147](#)
 - Print, [1146](#)
 - Selection, [1147](#)
 - SelectionMap, [1145](#)
 - SetSortFunction, [1146](#)
 - SetTagsToRead, [1146](#)
 - Sort, [1147](#)
 - Sorter, [1145](#)
 - SortFunc, [1148](#)
 - SortFunction, [1145](#)
 - StableSort, [1147](#)
 - TagsToRead, [1148](#)
- gdcm::Spacing, [1148](#)
 - ~Spacing, [1150](#)
 - CALIBRATED, [1150](#)
 - ComputePixelAspectRatioFromPixelSpacing, [1150](#)
 - DETECTOR, [1150](#)
 - MAGNIFIED, [1150](#)
 - Spacing, [1150](#)
 - SpacingType, [1150](#)
 - UNKNOWN, [1150](#)
- gdcm::Spectroscopy, [1150](#)
 - Spectroscopy, [1151](#)
- gdcm::SplitMosaicFilter, [1151](#)

- ~SplitMosaicFilter, [1152](#)
- ComputeCSAImageHeaderInfo, [1152](#)
- ComputeCSASeriesHeaderInfo, [1152](#)
- ComputeMOSAICDimensions, [1153](#)
- ComputeMOSAICImagePositionPatient, [1153](#)
- ComputeMOSAICSliceNormal, [1153](#)
- ComputeMOSAICSlicePosition, [1153](#)
- GetAcquisitionSize, [1153](#)
- GetFile, [1154](#)
- GetImage, [1154](#)
- GetNumberOfImagesInMosaic, [1154](#)
- SetFile, [1154](#)
- SetImage, [1154](#)
- Split, [1154](#)
- SplitMosaicFilter, [1152](#)
- gdcm::StartEvent, [1155](#)
- gdcm::static_assert_test< x >, [1156](#)
- gdcm::STATIC_ASSERTION_FAILURE< true >, [1157](#)
- value, [1158](#)
- gdcm::STATIC_ASSERTION_FAILURE< x >, [1157](#)
- gdcm::StreamImageReader, [1158](#)
- ~StreamImageReader, [1159](#)
- CanReadImage, [1160](#)
- DefinePixelExtent, [1160](#)
- DefineProperBufferLength, [1160](#)
- GetDimensionsValueForResolution, [1160](#)
- GetFile, [1161](#)
- Read, [1161](#)
- ReadImageInformation, [1161](#)
- SetFileName, [1161](#)
- SetStream, [1162](#)
- StreamImageReader, [1159](#)
- gdcm::StreamImageWriter, [1162](#)
- ~StreamImageWriter, [1164](#)
- CanWriteFile, [1165](#)
- DefinePixelExtent, [1165](#)
- DefineProperBufferLength, [1165](#)
- mElementOffsets, [1167](#)
- mElementOffsets1, [1167](#)
- mSPFile, [1167](#)
- mWriter, [1168](#)
- mXMax, [1168](#)
- mXMin, [1168](#)
- mYMax, [1168](#)
- mYMin, [1168](#)
- mZMax, [1168](#)
- mZMin, [1168](#)
- SetFile, [1165](#)
- SetFileName, [1166](#)
- SetStream, [1166](#)
- StreamImageWriter, [1164](#)
- Write, [1166](#)
- WriteImageInformation, [1166](#)
- WriteImageSubregionRAW, [1167](#)
- WriteRawHeader, [1167](#)
- gdcm::StrictScanner, [1169](#)
- ~StrictScanner, [1173](#)
- AddPrivateTag, [1173](#)
- AddSkipTag, [1173](#)
- AddTag, [1174](#)
- Begin, [1174](#)
- ClearSkipTags, [1174](#)
- ClearTags, [1174](#)
- ConstIterator, [1172](#)
- End, [1174](#)
- GetAllFileNamesFromTagToValue, [1174](#)
- GetFilenameFromTagToValue, [1174](#)
- GetFileNames, [1175](#)
- GetKeys, [1175](#)
- GetMapping, [1175](#)
- GetMappingFromTagToValue, [1175](#)
- GetMappings, [1175](#)
- GetOrderedValues, [1175](#)
- GetValue, [1176](#)
- GetValues, [1176](#)
- IsKey, [1176](#)
- MappingType, [1172](#)
- New, [1176](#)
- operator<<, [1178](#)
- Print, [1177](#)
- PrintTable, [1177](#)
- ProcessPublicTag, [1177](#)
- Scan, [1177](#)
- StrictScanner, [1173](#)
- TagToValue, [1172](#)
- TagToValueValueType, [1173](#)
- ValueType, [1173](#)
- gdcm::StrictScanner2, [1178](#)
- ~StrictScanner2, [1183](#)
- AddPrivateTag, [1183](#)
- AddPublicTag, [1183](#)
- AddSkipTag, [1183](#)
- Begin, [1183](#)
- ClearPrivateTags, [1183](#)
- ClearPublicTags, [1183](#)
- ClearSkipTags, [1184](#)
- End, [1184](#)
- GetAllFileNamesFromPrivateTagToValue, [1184](#)
- GetAllFileNamesFromPublicTagToValue, [1184](#)
- GetFilenameFromPrivateTagToValue, [1184](#)
- GetFilenameFromPublicTagToValue, [1184](#)
- GetFileNames, [1184](#)
- GetKeys, [1185](#)
- GetMappingFromPrivateTagToValue, [1185](#)
- GetMappingFromPublicTagToValue, [1185](#)
- GetPrivateMapping, [1185](#)

- GetPrivateMappings, 1185
- GetPrivateOrderedValues, 1185
- GetPrivateValue, 1185
- GetPrivateValues, 1186
- GetPublicMapping, 1186
- GetPublicMappings, 1186
- GetPublicOrderedValues, 1186
- GetPublicValue, 1186
- GetPublicValues, 1186
- GetValues, 1187
- IsKey, 1187
- New, 1187
- operator<<, 1188
- Print, 1187
- PrintTable, 1187
- PrivateBegin, 1187
- PrivateConstIterator, 1182
- PrivateEnd, 1188
- PrivateMappingType, 1182
- PrivateTagToValue, 1182
- PrivateTagToValueValueType, 1182
- ProcessPrivateTag, 1188
- ProcessPublicTag, 1188
- PublicConstIterator, 1182
- PublicMappingType, 1182
- PublicTagToValue, 1182
- PublicTagToValueValueType, 1182
- Scan, 1188
- StrictScanner2, 1183
- ValuesType, 1182
- gdcmm::StrictScanner2::ltstr, 785
 - operator(), 786
- gdcmm::StrictScanner::ltstr, 786
 - operator(), 786
- gdcmm::String< TDelimiter, TMaxLength, TPadChar
>, 1189
 - const_iterator, 1190
 - const_reference, 1190
 - const_reverse_iterator, 1191
 - difference_type, 1191
 - IsValid, 1192
 - iterator, 1191
 - operator const char *, 1192
 - pointer, 1191
 - reference, 1191
 - reverse_iterator, 1191
 - size_type, 1191
 - String, 1192
 - Trim, 1193
 - Truncate, 1193
 - value_type, 1191
- gdcmm::StringFilter, 1193
 - ~StringFilter, 1194
 - ExecuteQuery, 1195
 - FromString, 1195
 - GetFile, 1195
 - SetDicts, 1195
 - SetFile, 1195
 - StringFilter, 1194
 - ToString, 1196
 - ToStringPair, 1196, 1197
 - UseDictAlways, 1197
- gdcmm::Study, 1197
 - Study, 1197
- gdcmm::Subject, 1198
 - ~Subject, 1199
 - AddObserver, 1200
 - GetCommand, 1200
 - HasObserver, 1200
 - InvokeEvent, 1200
 - RemoveAllObservers, 1201
 - RemoveObserver, 1201
 - Subject, 1199
- gdcmm::Surface, 1201
 - ~Surface, 1205
 - GetAlgorithmFamily, 1205
 - GetAlgorithmName, 1205
 - GetAlgorithmVersion, 1206
 - GetAxisOfRotation, 1206
 - GetCenterOfRotation, 1206
 - GetFiniteVolume, 1206
 - GetManifold, 1206
 - GetMaximumPointDistance, 1206
 - GetMeanPointDistance, 1206
 - GetMeshPrimitive, 1206, 1207
 - GetNumberOfSurfacePoints, 1207
 - GetNumberOfVectors, 1207
 - GetPointCoordinatesData, 1207
 - GetPointPositionAccuracy, 1207
 - GetPointsBoundingBoxCoordinates, 1207
 - GetProcessingAlgorithm, 1207, 1208
 - GetRecommendedDisplayCIELabValue, 1208
 - GetRecommendedDisplayGrayscaleValue, 1208
 - GetRecommendedPresentationOpacity, 1208
 - GetRecommendedPresentationType, 1208
 - GetSTATES, 1208
 - GetSTATESString, 1208
 - GetSurfaceComments, 1208
 - GetSurfaceNumber, 1209
 - GetSurfaceProcessing, 1209
 - GetSurfaceProcessingDescription, 1209
 - GetSurfaceProcessingRatio, 1209
 - GetVectorAccuracy, 1209
 - GetVectorCoordinateData, 1209
 - GetVectorDimensionality, 1209
 - GetVIEWType, 1209
 - GetVIEWTypeString, 1209
 - NO, 1204

- POINTS, [1205](#)
- SetAlgorithmFamily, [1210](#)
- SetAlgorithmName, [1210](#)
- SetAlgorithmVersion, [1210](#)
- SetAxisOfRotation, [1210](#)
- SetCenterOfRotation, [1210](#)
- SetFiniteVolume, [1210](#)
- SetManifold, [1210](#)
- SetMaximumPointDistance, [1210](#)
- SetMeanPointDistance, [1211](#)
- SetMeshPrimitive, [1211](#)
- SetNumberOfSurfacePoints, [1211](#)
- SetNumberOfVectors, [1211](#)
- SetPointCoordinatesData, [1211](#)
- SetPointPositionAccuracy, [1211](#)
- SetPointsBoundingBoxCoordinates, [1211](#)
- SetProcessingAlgorithm, [1211](#)
- SetRecommendedDisplayCIELabValue, [1212](#)
- SetRecommendedDisplayGrayscaleValue, [1212](#)
- SetRecommendedPresentationOpacity, [1212](#)
- SetRecommendedPresentationType, [1212](#)
- SetSurfaceComments, [1212](#)
- SetSurfaceNumber, [1212](#)
- SetSurfaceProcessing, [1213](#)
- SetSurfaceProcessingDescription, [1213](#)
- SetSurfaceProcessingRatio, [1213](#)
- SetVectorAccuracy, [1213](#)
- SetVectorCoordinateData, [1213](#)
- SetVectorDimensionality, [1213](#)
- STATES, [1204](#)
- STATES_END, [1205](#)
- SURFACE, [1205](#)
- Surface, [1205](#)
- UNKNOWN, [1205](#)
- VIEWType, [1205](#)
- VIEWType_END, [1205](#)
- WIREFRAME, [1205](#)
- YES, [1204](#)
- gdcmm::SurfaceHelper, [1214](#)
 - ColorArray, [1214](#)
 - RecommendedDisplayCIELabToRGB, [1215](#)
 - RGBToRecommendedDisplayCIELab, [1216](#)
 - RGBToRecommendedDisplayGrayscale, [1216](#)
- gdcmm::SurfaceReader, [1217](#)
 - ~SurfaceReader, [1220](#)
 - GetNumberOfSurfaces, [1220](#)
 - Read, [1220](#)
 - ReadPointMacro, [1221](#)
 - ReadSurface, [1221](#)
 - ReadSurfaces, [1221](#)
 - SurfaceReader, [1220](#)
- gdcmm::SurfaceWriter, [1221](#)
 - ~SurfaceWriter, [1224](#)
 - ComputeNumberOfSurfaces, [1224](#)
 - GetNumberOfSurfaces, [1224](#)
 - NumberOfSurfaces, [1225](#)
 - PrepareWrite, [1225](#)
 - PrepareWritePointMacro, [1225](#)
 - SetNumberOfSurfaces, [1225](#)
 - SurfaceWriter, [1224](#)
 - Write, [1225](#)
- gdcmm::SwapCode, [1225](#)
 - BadBigEndian, [1227](#)
 - BadLittleEndian, [1227](#)
 - BigEndian, [1227](#)
 - GetIndex, [1227](#)
 - GetSwapCodeString, [1227](#)
 - LittleEndian, [1227](#)
 - operator SwapCode::SwapCodeType, [1227](#)
 - operator<<, [1228](#)
 - SwapCode, [1227](#)
 - SwapCodeType, [1226](#)
 - Unknown, [1226](#)
- gdcmm::SwapperDoOp, [1228](#)
 - Swap, [1228](#)
 - SwapArray, [1228](#)
- gdcmm::SwapperNoOp, [1229](#)
 - Swap, [1229](#)
 - SwapArray, [1229](#)
- gdcmm::System, [1229](#)
 - ConvertToUNC, [1231](#)
 - DeleteDirectory, [1231](#)
 - EncodeBytes, [1231](#)
 - FileExists, [1231](#)
 - FileIsDirectory, [1231](#)
 - FileIsSymlink, [1232](#)
 - FileSize, [1232](#)
 - FileTime, [1232](#)
 - FormatDateTime, [1232](#)
 - GetCurrentDateTime, [1233](#)
 - GetCurrentModuleFileName, [1233](#)
 - GetCurrentProcessFileName, [1233](#)
 - GetCurrentResourcesDirectory, [1233](#)
 - GetCWD, [1233](#)
 - GetHostName, [1234](#)
 - GetLastSystemError, [1234](#)
 - GetLocaleCharset, [1234](#)
 - GetPermissions, [1234](#)
 - GetTimezoneOffsetFromUTC, [1234](#)
 - MakeDirectory, [1234](#)
 - ParseDateTime, [1235](#)
 - RemoveFile, [1235](#)
 - SetPermissions, [1235](#)
 - StrCaseCmp, [1235](#)
 - StrNCaseCmp, [1235](#)
 - StrSep, [1236](#)
 - StrTokR, [1236](#)
- gdcmm::Table, [1236](#)

- ~Table, 1238
- GetTableEntry, 1238
- InsertEntry, 1238
- MapTableEntry, 1238
- operator<<, 1239
- operator=, 1239
- Table, 1238
- TableInternal, 1239
- gdcmm::TableEntry, 1239
 - ~TableEntry, 1240
 - TableEntry, 1240
- gdcmm::TableReader, 1240
 - ~TableReader, 1241
 - CharacterDataHandler, 1241
 - EndElement, 1241
 - GetDefs, 1242
 - GetFilename, 1242
 - HandleIOD, 1242
 - HandleIODEntry, 1242
 - HandleMacro, 1242
 - HandleMacroEntry, 1242
 - HandleMacroEntryDescription, 1242
 - HandleModule, 1242
 - HandleModuleEntry, 1243
 - HandleModuleEntryDescription, 1243
 - HandleModuleInclude, 1243
 - Read, 1243
 - SetFilename, 1243
 - StartElement, 1243
 - TableReader, 1241
- gdcmm::Tag, 1245
 - bytes, 1255
 - GetElement, 1248
 - GetElementTag, 1248
 - GetGroup, 1249
 - GetLength, 1249
 - GetPrivateCreator, 1249
 - IsGroupLength, 1249
 - IsGroupXX, 1249
 - IsIllegal, 1250
 - IsPrivate, 1250
 - IsPrivateCreator, 1250
 - IsPublic, 1250
 - operator!=, 1251
 - operator<, 1251
 - operator<<, 1255
 - operator<=, 1251
 - operator>>, 1255
 - operator=, 1251
 - operator==, 1251
 - operator[], 1251, 1252
 - PrintAsContinuousString, 1252
 - PrintAsContinuousUpperCaseString, 1252
 - PrintAsPipeSeparatedString, 1252
 - Read, 1252
 - ReadFromCommaSeparatedString, 1252
 - ReadFromContinuousString, 1253
 - ReadFromPipeSeparatedString, 1253
 - SetElement, 1253
 - SetElementTag, 1253, 1254
 - SetGroup, 1254
 - SetPrivateCreator, 1254
 - Tag, 1248
 - tag, 1255
 - tags, 1255
 - Write, 1254
- gdcmm::TagPath, 1255
 - ~TagPath, 1256
 - ConstructFromString, 1256
 - ConstructFromTagList, 1256
 - IsValid, 1257
 - Print, 1257
 - Push, 1257
 - TagPath, 1256
- gdcmm::terminal, 109
 - Attribute, 110
 - black, 110
 - blink, 110
 - blue, 110
 - bright, 110
 - Color, 110
 - CONSOLE, 110
 - cyan, 110
 - dim, 110
 - green, 110
 - hidden, 110
 - magenta, 110
 - Mode, 110
 - red, 110
 - reset, 110
 - reverse, 110
 - setAttribute, 111
 - setbgcolor, 111
 - setfgcolor, 111
 - setmode, 111
 - underline, 110
 - VT100, 110
 - white, 110
 - yellow, 110
- gdcmm::Testing, 1257
 - ~Testing, 1259
 - ComputeFileMD5, 1259
 - ComputeMD5, 1259
 - GetDataExtraRoot, 1259
 - GetDataRoot, 1260
 - GetFileName, 1260
 - GetFileNames, 1260
 - GetLossyFlagFromFile, 1260

- GetMD5DataImage, [1260](#)
- GetMD5DataImages, [1261](#)
- GetMD5FromBrokenFile, [1261](#)
- GetMD5FromFile, [1261](#)
- GetMediaStorageDataFile, [1261](#)
- GetMediaStorageDataFiles, [1261](#)
- GetMediaStorageFromFile, [1261](#)
- GetNumberOfFileNames, [1261](#)
- GetNumberOfMD5DataImages, [1262](#)
- GetNumberOfMediaStorageDataFiles, [1262](#)
- GetPixelSpacingDataRoot, [1262](#)
- GetSelectedPrivateGroupOffsetFromFile, [1262](#)
- GetSelectedTagsOffsetFromFile, [1262](#)
- GetSourceDirectory, [1262](#)
- GetStreamOffsetFromFile, [1262](#)
- GetTempDirectory, [1263](#)
- GetTempDirectoryW, [1263](#)
- GetTempFilename, [1263](#)
- GetTempFilenameW, [1263](#)
- MD5DataImagesType, [1259](#)
- MediaStorageDataFilesType, [1259](#)
- Print, [1263](#)
- Testing, [1259](#)
- gdcm::Trace, [1264](#)
 - ~Trace, [1265](#)
 - DebugOff, [1265](#)
 - DebugOn, [1265](#)
 - ErrorOff, [1266](#)
 - ErrorOn, [1266](#)
 - GetDebugFlag, [1266](#)
 - GetDebugStream, [1266](#)
 - GetErrorFlag, [1266](#)
 - GetErrorStream, [1266](#)
 - GetStream, [1266](#)
 - GetWarningFlag, [1266](#)
 - GetWarningStream, [1267](#)
 - SetDebug, [1267](#)
 - SetDebugStream, [1267](#)
 - SetError, [1267](#)
 - SetErrorStream, [1267](#)
 - SetStream, [1267](#)
 - SetStreamToFile, [1268](#)
 - SetWarning, [1268](#)
 - SetWarningStream, [1268](#)
 - Trace, [1265](#)
 - WarningOff, [1268](#)
 - WarningOn, [1268](#)
- gdcm::TransferSyntax, [1269](#)
 - CanStoreLossy, [1273](#)
 - CT_private_ELE, [1272](#)
 - DeflatedExplicitVRLittleEndian, [1271](#)
 - DeflatedImageFrameCompression, [1272](#)
 - Explicit, [1271](#)
 - ExplicitVRBigEndian, [1271](#)
 - ExplicitVRLittleEndian, [1271](#)
 - GetNegociatedType, [1273](#)
 - GetString, [1273](#)
 - GetSwapCode, [1273](#)
 - GetTSString, [1273](#)
 - GetTSType, [1273](#)
 - HTJ2K, [1272](#)
 - HTJ2KLossless, [1272](#)
 - HTJ2KRPCLLossless, [1272](#)
 - Implicit, [1271](#)
 - ImplicitVRBigEndianACRNEMA, [1272](#)
 - ImplicitVRBigEndianPrivateGE, [1271](#)
 - ImplicitVRLittleEndian, [1271](#)
 - IsEncapsulated, [1273](#)
 - IsEncoded, [1274](#)
 - IsExplicit, [1274](#)
 - IsImplicit, [1274](#)
 - IsLossless, [1274](#)
 - IsLossy, [1274](#)
 - IsValid, [1274](#)
 - JPEG2000, [1272](#)
 - JPEG2000Lossless, [1272](#)
 - JPEG2000Part2, [1272](#)
 - JPEG2000Part2Lossless, [1272](#)
 - JPEGBaselineProcess1, [1271](#)
 - JPEGExtendedProcess2_4, [1271](#)
 - JPEGExtendedProcess3_5, [1271](#)
 - JPEGFullProgressionProcess10_12, [1271](#)
 - JPEGLosslessProcess14, [1271](#)
 - JPEGLosslessProcess14_1, [1272](#)
 - JPEGLSLossless, [1272](#)
 - JPEGLSNearLossless, [1272](#)
 - JPEGSpectralSelectionProcess6_8, [1271](#)
 - JPIPRReferenced, [1272](#)
 - MPEG2MainProfile, [1272](#)
 - MPEG2MainProfileHighLevel, [1272](#)
 - MPEG4AVCH264BDcompatibleHighProfileLevel4_1, [1272](#)
 - MPEG4AVCH264HighProfileLevel4_1, [1272](#)
 - NegotiatedType, [1271](#)
 - operator TSType, [1274](#)
 - operator<<, [1275](#)
 - RLELossless, [1272](#)
 - TransferSyntax, [1272](#)
 - TS_END, [1272](#)
 - TSType, [1271](#)
 - Unknown, [1271](#)
 - WeirdPapryus, [1272](#)
- gdcm::Type, [1279](#)
 - GetTypeString, [1281](#)
 - GetTypeType, [1281](#)
 - operator TypeType, [1281](#)
 - operator<<, [1281](#)
 - T1, [1280](#)

- T1C, [1280](#)
- T2, [1280](#)
- T2C, [1280](#)
- T3, [1280](#)
- Type, [1280](#)
- TypeType, [1280](#)
- UNKNOWN, [1280](#)
- gdcm::UI, [1281](#)
 - Internal, [1282](#)
 - operator<<, [1282](#)
- gdcm::UIDGenerator, [1282](#)
 - Generate, [1283](#)
 - GenerateUUID, [1283](#)
 - GetGDCMUID, [1283](#)
 - GetRoot, [1284](#)
 - IsValid, [1284](#)
 - SetRoot, [1284](#)
 - UIDGenerator, [1283](#)
- gdcm::UIDs, [1285](#)
 - AbstractMultiDimensionalImageModel, [1311](#)
 - AcquisitionContextSRStorage, [1310](#)
 - AdultMouseAnatomyOntology, [1308](#)
 - AdvancedBlendingPresentationStateStorage, [1309](#)
 - AmbulatoryECGWaveformStorage, [1304](#)
 - ArterialPulseWaveformStorage, [1309](#)
 - AudioSRStorageTrialRetired, [1305](#)
 - AutorefractiveMeasurementsStorage, [1309](#)
 - BasicAnnotationBoxSOPClass, [1303](#)
 - BasicColorImageBoxSOPClass, [1303](#)
 - BasicColorPrintManagementMetaSOPClass, [1303](#)
 - BasicFilmBoxSOPClass, [1303](#)
 - BasicFilmSessionSOPClass, [1303](#)
 - BasicGrayscaleImageBoxSOPClass, [1303](#)
 - BasicGrayscalePrintManagementMetaSOPClass, [1303](#)
 - BasicPrintImageOverlayBoxSOPClassRetired, [1303](#)
 - BasicStructuredDisplayStorage, [1310](#)
 - BasicStudyContentNotificationSOPClassRetired, [1302](#)
 - BasicTextSRStorage, [1305](#)
 - BasicVoiceAudioWaveformStorage, [1304](#)
 - BlendingSoftcopyPresentationStateStorageSOPClass, [1305](#)
 - BreastImagingRelevantPatientInformationQuery, [1307](#)
 - BreastProjectionXRayImageStorageForPresentation, [1309](#)
 - BreastProjectionXRayImageStorageForProcessing, [1309](#)
 - BreastTomosynthesisImageStorage, [1308](#)
 - CardiacElectrophysiologyWaveformStorage, [1304](#)
 - CardiacRelevantPatientInformationQuery, [1307](#)
 - ChestCADSRStorage, [1306](#)
 - ColonCADSRStorage, [1310](#)
 - ColorPaletteQueryRetrieveInformationModelFIND, [1311](#)
 - ColorPaletteQueryRetrieveInformationModelGET, [1311](#)
 - ColorPaletteQueryRetrieveInformationModelMOVE, [1311](#)
 - ColorPaletteStorage, [1311](#)
 - ColorSoftcopyPresentationStateStorageSOPClass, [1304](#)
 - CompositeInstanceRetrieveWithoutBulkDataGET, [1310](#)
 - CompositeInstanceRootRetrieveGET, [1310](#)
 - CompositeInstanceRootRetrieveMOVE, [1310](#)
 - CompositingPlanarMPRVolumetricPresentationStateStorage, [1309](#)
 - Comprehensive3DSRStorage, [1310](#)
 - ComprehensiveSRStorage, [1305](#)
 - ComprehensiveSRStorageTrialRetired, [1305](#)
 - ComputedRadiographyImageStorage, [1304](#)
 - ContentAssessmentResultsStorage, [1310](#)
 - CornealTopographyMapStorage, [1310](#)
 - CTDefinedProcedureProtocolStorage, [1310](#)
 - CTImageStorage, [1304](#)
 - CTPerformedProcedureProtocolStorage, [1310](#)
 - DefinedProcedureProtocolInformationModelFIND, [1311](#)
 - DefinedProcedureProtocolInformationModelGET, [1311](#)
 - DefinedProcedureProtocolInformationModelMOVE, [1311](#)
 - DeflatedExplicitVRLittleEndian, [1301](#)
 - DeformableSpatialRegistrationStorage, [1305](#)
 - DetachedInterpretationManagementSOPClassRetired, [1303](#)
 - DetachedPatientManagementMetaSOPClassRetired, [1303](#)
 - DetachedPatientManagementSOPClassRetired, [1303](#)
 - DetachedResultsManagementMetaSOPClassRetired, [1303](#)
 - DetachedResultsManagementSOPClassRetired, [1303](#)
 - DetachedStudyManagementMetaSOPClassRetired, [1303](#)
 - DetachedStudyManagementSOPClassRetired, [1303](#)
 - DetachedVisitManagementSOPClassRetired, [1303](#)
 - DetailSRStorageTrialRetired, [1305](#)

- dicomAETitle, [1307](#)
- dicomApplicationCluster, [1307](#)
- DICOMApplicationContextName, [1303](#)
- dicomAssociationAcceptor, [1307](#)
- dicomAssociationInitiator, [1307](#)
- dicomAuthorizedNodeCertificateReference, [1307](#)
- dicomConfigurationRoot, [1308](#)
- DICOMContentMappingResource, [1311](#)
- DICOMControlledTerminology, [1303](#)
- dicomDescription, [1307](#)
- dicomDevice, [1308](#)
- dicomDeviceName, [1307](#)
- dicomDeviceSerialNumber, [1307](#)
- dicomDevicesRoot, [1308](#)
- dicomHostname, [1307](#)
- dicomInstalled, [1307](#)
- dicomInstitutionAddress, [1307](#)
- dicomInstitutionDepartmentName, [1307](#)
- dicomInstitutionName, [1307](#)
- dicomIssuerOfPatientID, [1307](#)
- dicomManufacturer, [1307](#)
- dicomManufacturerModelName, [1307](#)
- dicomNetworkAE, [1308](#)
- dicomNetworkConnection, [1308](#)
- dicomNetworkConnectionReference, [1307](#)
- dicomPort, [1307](#)
- dicomPreferredCalledAETitle, [1307](#)
- dicomPreferredCallingAETitle, [1307](#)
- dicomPrimaryDeviceType, [1307](#)
- dicomRelatedDeviceReference, [1307](#)
- dicomSoftwareVersion, [1307](#)
- dicomSOPClass, [1307](#)
- dicomStationName, [1307](#)
- dicomSupportedCharacterSet, [1307](#)
- dicomThisNodeCertificateReference, [1307](#)
- dicomTLSCyphersuite, [1307](#)
- dicomTransferCapability, [1308](#)
- dicomTransferRole, [1307](#)
- dicomTransferSyntax, [1307](#)
- DICOMUIDRegistry, [1303](#)
- dicomUniqueAETitle, [1308](#)
- dicomUniqueAETitlesRegistryRoot, [1308](#)
- dicomVendorData, [1307](#)
- DICOS2DAITStorage, [1310](#)
- DICOS3DAITStorage, [1310](#)
- DICOSCTImageStorage, [1310](#)
- DICOSDigitalXRayImageStorageForPresentation, [1310](#)
- DICOSDigitalXRayImageStorageForProcessing, [1310](#)
- DICOSQuadrupoleResonanceQRStorage, [1310](#)
- DICOSThreatDetectionReportStorage, [1310](#)
- DigitalIntraoralXRayImageStorageForPresentation, [1304](#)
- DigitalIntraoralXRayImageStorageForProcessing, [1304](#)
- DigitalMammographyXRayImageStorageForPresentation, [1304](#)
- DigitalMammographyXRayImageStorageForProcessing, [1304](#)
- DigitalXRayImageStorageForPresentation, [1304](#)
- DigitalXRayImageStorageForProcessing, [1304](#)
- DisplaySystemSOPClass, [1308](#)
- DisplaySystemSOPInstance, [1308](#)
- ECG12leadWaveformStorage, [1304](#)
- EddyCurrentImageStorage, [1310](#)
- EddyCurrentMultiframeImageStorage, [1310](#)
- EncapsulatedCDASTorage, [1306](#)
- EncapsulatedPDFStorage, [1306](#)
- EncapsulatedSTLStorage, [1310](#)
- EnhancedCTImageStorage, [1304](#)
- EnhancedMRColorImageStorage, [1311](#)
- EnhancedMRImageStorage, [1304](#)
- EnhancedPETImageStorage, [1310](#)
- EnhancedSRStorage, [1305](#)
- EnhancedUSVolumeStorage, [1308](#)
- EnhancedXAImageStorage, [1305](#)
- EnhancedXRFImageStorage, [1305](#)
- ExplicitVRBigEndian, [1301](#)
- ExplicitVRLittleEndian, [1301](#)
- ExtensibleSRStorage, [1310](#)
- FallColorPaletteSOPInstance, [1308](#)
- GeneralAudioWaveformStorage, [1309](#)
- GeneralECGWaveformStorage, [1304](#)
- GeneralPurposePerformedProcedureStepSOPClass, [1306](#)
- GeneralPurposeScheduledProcedureStepSOPClass, [1306](#)
- GeneralPurposeWorklistInformationModelFIND, [1306](#)
- GeneralPurposeWorklistManagementMetaSOPClass, [1306](#)
- GeneralRelevantPatientInformationQuery, [1307](#)
- GenericImplantTemplateInformationModelFIND, [1311](#)
- GenericImplantTemplateInformationModelGET, [1311](#)
- GenericImplantTemplateInformationModelMOVE, [1311](#)
- GenericImplantTemplateStorage, [1311](#)
- GetName, [1327](#)
- GetNumberOfTransferSyntaxStrings, [1327](#)
- GetString, [1328](#)
- GetTransferSyntaxString, [1328](#)
- GetTransferSyntaxStrings, [1328](#)
- GetUIDName, [1328](#)
- GetUIDString, [1328](#)

- GrayscalePlanarMPRVolumetricPresentation-
StateStorage, [1309](#)
- GrayscaleSoftcopyPresentationStateStorage-
SOPClass, [1304](#)
- HangingProtocolInformationModelFIND, [1307](#)
- HangingProtocolInformationModelGET, [1311](#)
- HangingProtocolInformationModelMOVE, [1307](#)
- HangingProtocolStorage, [1307](#)
- HardcopyColorImageStorageSOPClassRetired,
[1304](#)
- HardcopyGrayscaleImageStorageSOPClassRe-
tired, [1303](#)
- HemodynamicWaveformStorage, [1304](#)
- HEVCH_265Main10ProfileLevel5_1, [1309](#)
- HEVCH_265MainProfileLevel5_1, [1309](#)
- HotIronColorPaletteSOPInstance, [1309](#)
- HotMetalBlueColorPaletteSOPInstance, [1308](#)
- ICBM452T1FrameofReference, [1302](#)
- ICBMSingleSubjectMRIFrameofReference, [1302](#)
- ICD11, [1308](#)
- ImageBiomarkerStandardisationInitiative, [1308](#)
- ImageOverlayBoxSOPClassRetired, [1303](#)
- ImplantAssemblyTemplateInformationMod-
elFIND, [1311](#)
- ImplantAssemblyTemplateInformationModel-
GET, [1311](#)
- ImplantAssemblyTemplateInformationModel-
MOVE, [1311](#)
- ImplantAssemblyTemplateStorage, [1311](#)
- ImplantationPlanSRStorage, [1310](#)
- ImplantTemplateGroupInformationModelFIND,
[1311](#)
- ImplantTemplateGroupInformationModelGET,
[1311](#)
- ImplantTemplateGroupInformationModel-
MOVE, [1311](#)
- ImplantTemplateGroupStorage, [1311](#)
- ImplicitVRLittleEndianDefaultTransferSyntax-
forDICOM, [1301](#)
- InstanceAvailabilityNotificationSOPClass, [1306](#)
- IntegratedTaxonomicInformationSystemITIS-
TaxonomicSerialNumberTSN, [1308](#)
- IntraocularLensCalculationsStorage, [1310](#)
- IntravascularOpticalCoherenceTomographyIm-
ageStorageForPresentation, [1309](#)
- IntravascularOpticalCoherenceTomographyIm-
ageStorageForProcessing, [1309](#)
- JPEG2000ImageCompression, [1302](#)
- JPEG2000ImageCompressionLosslessOnly, [1301](#)
- JPEG2000Part2MulticomponentImageCom-
pression, [1302](#)
- JPEG2000Part2MulticomponentImageCom-
pressionLosslessOnly, [1302](#)
- JPEGBaselineProcess1DefaultTransferSyntax-
forLossyJPEG8BitImageCompression, [1301](#)
- JPEGExtendedHierarchicalProcess1618Retired,
[1301](#)
- JPEGExtendedHierarchicalProcess1719Retired,
[1301](#)
- JPEGExtendedProcess24DefaultTransferSyn-
taxforLossyJPEG12BitImageCompression-
Process4only, [1301](#)
- JPEGExtendedProcess35Retired, [1301](#)
- JPEGFullProgressionHierarchicalProcess2426Re-
tired, [1301](#)
- JPEGFullProgressionHierarchicalProcess2527Re-
tired, [1301](#)
- JPEGFullProgressionNonHierarchicalPro-
cess1012Retired, [1301](#)
- JPEGFullProgressionNonHierarchicalPro-
cess1113Retired, [1301](#)
- JPEGLosslessHierarchicalProcess28Retired,
[1301](#)
- JPEGLosslessHierarchicalProcess29Retired,
[1301](#)
- JPEGLosslessNonHierarchicalFirstOrderPredic-
tionProcess14SelectionValue1DefaultTrans-
ferSyntaxforLosslessJPEGImageCompres-
sion, [1301](#)
- JPEGLosslessNonHierarchicalProcess14, [1301](#)
- JPEGLosslessNonHierarchicalProcess15Retired,
[1301](#)
- JPEGLSLosslessImageCompression, [1301](#)
- JPEGLSLossyNearLosslessImageCompression,
[1301](#)
- JPEGSpectralSelectionHierarchicalProcess2022Re-
tired, [1301](#)
- JPEGSpectralSelectionHierarchicalProcess2123Re-
tired, [1301](#)
- JPEGSpectralSelectionNonHierarchicalPro-
cess68Retired, [1301](#)
- JPEGSpectralSelectionNonHierarchicalPro-
cess79Retired, [1301](#)
- JPIPReferenced, [1302](#)
- JPIPReferencedDeflate, [1302](#)
- KeratometryMeasurementsStorage, [1309](#)
- KeyObjectSelectionDocumentStorage, [1305](#)
- LegacyConvertedEnhancedCTImageStorage,
[1308](#)
- LegacyConvertedEnhancedMRIImageStorage,
[1308](#)
- LegacyConvertedEnhancedPETImageStorage,
[1308](#)
- LensometryMeasurementsStorage, [1309](#)
- MacularGridThicknessandVolumeReportStor-
age, [1310](#)
- MammographyCADSRStorage, [1305](#)
- MayoClinicNonradiologicalImagesSBSAnatomi-

- calSurfaceRegionGuide, [1308](#)
- MediaCreationManagementSOPClassUID, [1304](#)
- MediaStorageDirectoryStorage, [1302](#)
- ModalityPerformedProcedureStepNotification-SOPClass, [1303](#)
- ModalityPerformedProcedureStepRetrieveSOP-Class, [1303](#)
- ModalityPerformedProcedureStepSOPClass, [1303](#)
- ModalityWorklistInformationModelFIND, [1306](#)
- MouseGenomeInitiativeMGI, [1308](#)
- MPEG2MainProfileHighLevel, [1308](#)
- MPEG2MainProfileMainLevel, [1302](#)
- MPEG4AVCH_264BDcompatibleHighProfileLevel4_1, [1308](#)
- MPEG4AVCH_264HighProfileLevel4_1, [1308](#)
- MPEG4AVCH_264HighProfileLevel4_2For2DVideo, [1309](#)
- MPEG4AVCH_264HighProfileLevel4_2For3DVideo, [1309](#)
- MPEG4AVCH_264StereoHighProfileLevel4_2, [1309](#)
- MRImageStorage, [1304](#)
- MRSpectroscopyStorage, [1304](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, [1304](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, [1304](#)
- MultiframeSingleBitSecondaryCaptureImageStorage, [1304](#)
- MultiframeTrueColorSecondaryCaptureImageStorage, [1304](#)
- MultipleVolumeRenderingVolumetricPresentationStateStorage, [1309](#)
- NativeDICOMModel, [1311](#)
- NewYorkUniversityMelanomaClinicalCooperativeGroup, [1308](#)
- NuclearMedicineImageStorage, [1305](#)
- NuclearMedicineImageStorageRetired, [1304](#)
- Null0, [1309](#)
- Null1, [1309](#)
- operator TSType, [1328](#)
- OphthalmicAxialMeasurementsStorage, [1309](#)
- OphthalmicOpticalCoherenceTomographyB-scanVolumeAnalysisStorage, [1309](#)
- OphthalmicOpticalCoherenceTomographyEnFaceImageStorage, [1309](#)
- OphthalmicPhotography16BitImageStorage, [1305](#)
- OphthalmicPhotography8BitImageStorage, [1305](#)
- OphthalmicThicknessMapStorage, [1310](#)
- OphthalmicTomographyImageStorage, [1305](#)
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage, [1310](#)
- Papyrus3ImplicitVRLittleEndian, [1308](#)
- ParametricMapStorage, [1309](#)
- PatientRadiationDoseSRStorage, [1310](#)
- PatientRootQueryRetrieveInformationModelFIND, [1306](#)
- PatientRootQueryRetrieveInformationModelGET, [1306](#)
- PatientRootQueryRetrieveInformationModelMOVE, [1306](#)
- PatientStudyOnlyQueryRetrieveInformationModelFINDRetired, [1306](#)
- PatientStudyOnlyQueryRetrieveInformationModelGETRetired, [1306](#)
- PatientStudyOnlyQueryRetrieveInformationModelMOVERetired, [1306](#)
- PerformedImagingAgentAdministrationSRStorage, [1310](#)
- PET20StepColorPaletteSOPInstance, [1308](#)
- PETColorPaletteSOPInstance, [1308](#)
- PlannedImagingAgentAdministrationSRStorage, [1310](#)
- PositronEmissionTomographyImageStorage, [1306](#)
- PresentationLUTSOPClass, [1303](#)
- PrinterConfigurationRetrievalSOPClass, [1303](#)
- PrinterConfigurationRetrievalSOPInstance, [1303](#)
- PrinterSOPClass, [1303](#)
- PrinterSOPInstance, [1303](#)
- PrintJobSOPClass, [1303](#)
- PrintQueueManagementSOPClassRetired, [1303](#)
- PrintQueueSOPInstanceRetired, [1303](#)
- ProceduralEventLoggingSOPClass, [1302](#)
- ProceduralEventLoggingSOPInstance, [1302](#)
- ProcedureLogStorage, [1305](#)
- ProductCharacteristicsQuerySOPClass, [1307](#)
- ProtocolApprovalInformationModelFIND, [1310](#)
- ProtocolApprovalInformationModelGET, [1310](#)
- ProtocolApprovalInformationModelMOVE, [1310](#)
- ProtocolApprovalStorage, [1310](#)
- PseudoColorSoftcopyPresentationStateStorage-SOPClass, [1305](#)
- PubChemCompoundCID, [1308](#)
- PullPrintRequestSOPClassRetired, [1304](#)
- PullStoredPrintManagementMetaSOPClassRetired, [1304](#)
- RadiomicsOntology, [1308](#)
- RadiopharmaceuticalRadiationDoseSRStorage, [1310](#)
- RawDataStorage, [1305](#)
- RealWorldValueMappingStorage, [1305](#)
- ReferencedColorPrintManagementMetaSOP-ClassRetired, [1303](#)

- ReferencedGrayscalePrintManagementMeta-SOPClassRetired, [1303](#)
 ReferencedImageBoxSOPClassRetired, [1303](#)
 RespiratoryWaveformStorage, [1309](#)
 RFC2557MIMEencapsulation, [1302](#)
 RLELossless, [1302](#)
 RTBeamsDeliveryInstructionStorage, [1311](#)
 RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft, [1306](#)
 RTBeamsTreatmentRecordStorage, [1306](#)
 RTBrachyApplicationSetupDeliveryInstructionStorage, [1311](#)
 RTBrachyTreatmentRecordStorage, [1306](#)
 RTConventionalMachineVerification, [1311](#)
 RTConventionalMachineVerificationSupplement74FrozenDraft, [1306](#)
 RTDoseStorage, [1306](#)
 RTImageStorage, [1306](#)
 RTIonBeamsTreatmentRecordStorage, [1306](#)
 RTIonMachineVerification, [1311](#)
 RTIonMachineVerificationSupplement74FrozenDraft, [1306](#)
 RTIonPlanStorage, [1306](#)
 RTPhysicianIntentStorage, [1310](#)
 RTPlanStorage, [1306](#)
 RTSegmentAnnotationStorage, [1310](#)
 RTStructureSetStorage, [1306](#)
 RTTreatmentSummaryRecordStorage, [1306](#)
 SecondaryCaptureImageStorage, [1304](#)
 SegmentationStorage, [1305](#)
 SegmentedVolumeRenderingVolumetricPresentationStateStorage, [1309](#)
 SetFromUID, [1328](#)
 SimplifiedAdultEchoSRStorage, [1310](#)
 SpatialFiducialsStorage, [1305](#)
 SpatialRegistrationStorage, [1305](#)
 SpectaclePrescriptionReportStorage, [1309](#)
 SPM2AVG152PDFFrameofReference, [1302](#)
 SPM2AVG152T1FrameofReference, [1302](#)
 SPM2AVG152T2FrameofReference, [1302](#)
 SPM2AVG305T1FrameofReference, [1302](#)
 SPM2BRAINMASKFrameofReference, [1302](#)
 SPM2CSFFFrameofReference, [1302](#)
 SPM2EPIFrameofReference, [1302](#)
 SPM2FILT1FrameofReference, [1302](#)
 SPM2GRAYFrameofReference, [1302](#)
 SPM2PDFFrameofReference, [1302](#)
 SPM2PETFrameofReference, [1302](#)
 SPM2SINGLESUBJT1FrameofReference, [1302](#)
 SPM2SPECTFrameofReference, [1302](#)
 SPM2T1FrameofReference, [1302](#)
 SPM2T2FrameofReference, [1302](#)
 SPM2TRANSMFrameofReference, [1302](#)
 SPM2WHITEFrameofReference, [1302](#)
 SpringColorPaletteSOPInstance, [1308](#)
 StandaloneCurveStorageRetired, [1304](#)
 StandaloneModalityLUTStorageRetired, [1304](#)
 StandaloneOverlayStorageRetired, [1304](#)
 StandalonePETCurveStorageRetired, [1306](#)
 StandaloneVOILUTStorageRetired, [1304](#)
 StereometricRelationshipStorage, [1305](#)
 StorageCommitmentPullModelSOPClassRetired, [1302](#)
 StorageCommitmentPullModelSOPInstanceRetired, [1302](#)
 StorageCommitmentPushModelSOPClass, [1302](#)
 StorageCommitmentPushModelSOPInstance, [1302](#)
 StorageServiceClass, [1303](#)
 StoredPrintStorageSOPClassRetired, [1303](#)
 StudyComponentManagementSOPClassRetired, [1303](#)
 StudyRootQueryRetrieveInformationModelFIND, [1306](#)
 StudyRootQueryRetrieveInformationModelGET, [1306](#)
 StudyRootQueryRetrieveInformationModelMOVE, [1306](#)
 SubjectiveRefractionMeasurementsStorage, [1309](#)
 SubstanceAdministrationLoggingSOPClass, [1302](#)
 SubstanceAdministrationLoggingSOPInstance, [1302](#)
 SubstanceApprovalQuerySOPClass, [1307](#)
 SummerColorPaletteSOPInstance, [1308](#)
 SurfaceScanMeshStorage, [1309](#)
 SurfaceScanPointCloudStorage, [1309](#)
 SurfaceSegmentationStorage, [1308](#)
 TalairachBrainAtlasFrameofReference, [1302](#)
 TextSRStorageTrialRetired, [1305](#)
 TractographyResultsStorage, [1309](#)
 TransferSyntaxStringsType, [1301](#)
 TSName, [1301](#)
 TSType, [1311](#)
 UberonOntology, [1308](#)
 uid_1_2_840_10008_15_0_3_1, [1320](#)
 uid_1_2_840_10008_15_0_3_10, [1321](#)
 uid_1_2_840_10008_15_0_3_11, [1321](#)
 uid_1_2_840_10008_15_0_3_12, [1321](#)
 uid_1_2_840_10008_15_0_3_13, [1321](#)
 uid_1_2_840_10008_15_0_3_14, [1321](#)
 uid_1_2_840_10008_15_0_3_15, [1321](#)
 uid_1_2_840_10008_15_0_3_16, [1321](#)
 uid_1_2_840_10008_15_0_3_17, [1321](#)
 uid_1_2_840_10008_15_0_3_18, [1321](#)
 uid_1_2_840_10008_15_0_3_19, [1321](#)
 uid_1_2_840_10008_15_0_3_2, [1320](#)
 uid_1_2_840_10008_15_0_3_20, [1321](#)

uid_1_2_840_10008_15_0_3_21, [1321](#)
uid_1_2_840_10008_15_0_3_22, [1321](#)
uid_1_2_840_10008_15_0_3_23, [1321](#)
uid_1_2_840_10008_15_0_3_24, [1321](#)
uid_1_2_840_10008_15_0_3_25, [1321](#)
uid_1_2_840_10008_15_0_3_26, [1321](#)
uid_1_2_840_10008_15_0_3_27, [1321](#)
uid_1_2_840_10008_15_0_3_28, [1321](#)
uid_1_2_840_10008_15_0_3_29, [1321](#)
uid_1_2_840_10008_15_0_3_3, [1320](#)
uid_1_2_840_10008_15_0_3_30, [1321](#)
uid_1_2_840_10008_15_0_3_31, [1321](#)
uid_1_2_840_10008_15_0_3_4, [1320](#)
uid_1_2_840_10008_15_0_3_5, [1320](#)
uid_1_2_840_10008_15_0_3_6, [1320](#)
uid_1_2_840_10008_15_0_3_7, [1321](#)
uid_1_2_840_10008_15_0_3_8, [1321](#)
uid_1_2_840_10008_15_0_3_9, [1321](#)
uid_1_2_840_10008_15_0_4_1, [1321](#)
uid_1_2_840_10008_15_0_4_2, [1321](#)
uid_1_2_840_10008_15_0_4_3, [1321](#)
uid_1_2_840_10008_15_0_4_4, [1321](#)
uid_1_2_840_10008_15_0_4_5, [1321](#)
uid_1_2_840_10008_15_0_4_6, [1321](#)
uid_1_2_840_10008_15_0_4_7, [1321](#)
uid_1_2_840_10008_15_0_4_8, [1322](#)
uid_1_2_840_10008_15_1_1, [1327](#)
uid_1_2_840_10008_1_1, [1312](#)
uid_1_2_840_10008_1_2, [1312](#)
uid_1_2_840_10008_1_20, [1322](#)
uid_1_2_840_10008_1_20_1, [1313](#)
uid_1_2_840_10008_1_20_1_1, [1313](#)
uid_1_2_840_10008_1_20_2, [1313](#)
uid_1_2_840_10008_1_20_2_1, [1313](#)
uid_1_2_840_10008_1_2_1, [1312](#)
uid_1_2_840_10008_1_2_1_99, [1312](#)
uid_1_2_840_10008_1_2_2, [1312](#)
uid_1_2_840_10008_1_2_4_100, [1312](#)
uid_1_2_840_10008_1_2_4_101, [1322](#)
uid_1_2_840_10008_1_2_4_102, [1322](#)
uid_1_2_840_10008_1_2_4_103, [1322](#)
uid_1_2_840_10008_1_2_4_104, [1323](#)
uid_1_2_840_10008_1_2_4_105, [1323](#)
uid_1_2_840_10008_1_2_4_106, [1323](#)
uid_1_2_840_10008_1_2_4_107, [1323](#)
uid_1_2_840_10008_1_2_4_108, [1323](#)
uid_1_2_840_10008_1_2_4_50, [1312](#)
uid_1_2_840_10008_1_2_4_51, [1312](#)
uid_1_2_840_10008_1_2_4_52, [1312](#)
uid_1_2_840_10008_1_2_4_53, [1312](#)
uid_1_2_840_10008_1_2_4_54, [1312](#)
uid_1_2_840_10008_1_2_4_55, [1312](#)
uid_1_2_840_10008_1_2_4_56, [1312](#)
uid_1_2_840_10008_1_2_4_57, [1312](#)
uid_1_2_840_10008_1_2_4_58, [1312](#)
uid_1_2_840_10008_1_2_4_59, [1312](#)
uid_1_2_840_10008_1_2_4_60, [1312](#)
uid_1_2_840_10008_1_2_4_61, [1312](#)
uid_1_2_840_10008_1_2_4_62, [1312](#)
uid_1_2_840_10008_1_2_4_63, [1312](#)
uid_1_2_840_10008_1_2_4_64, [1312](#)
uid_1_2_840_10008_1_2_4_65, [1312](#)
uid_1_2_840_10008_1_2_4_66, [1312](#)
uid_1_2_840_10008_1_2_4_70, [1312](#)
uid_1_2_840_10008_1_2_4_80, [1312](#)
uid_1_2_840_10008_1_2_4_81, [1312](#)
uid_1_2_840_10008_1_2_4_90, [1312](#)
uid_1_2_840_10008_1_2_4_91, [1312](#)
uid_1_2_840_10008_1_2_4_92, [1312](#)
uid_1_2_840_10008_1_2_4_93, [1312](#)
uid_1_2_840_10008_1_2_4_94, [1312](#)
uid_1_2_840_10008_1_2_4_95, [1312](#)
uid_1_2_840_10008_1_2_5, [1312](#)
uid_1_2_840_10008_1_2_6_1, [1312](#)
uid_1_2_840_10008_1_2_6_2, [1313](#)
uid_1_2_840_10008_1_3_10, [1313](#)
uid_1_2_840_10008_1_40, [1314](#)
uid_1_2_840_10008_1_40_1, [1314](#)
uid_1_2_840_10008_1_42, [1314](#)
uid_1_2_840_10008_1_42_1, [1314](#)
uid_1_2_840_10008_1_4_1_1, [1313](#)
uid_1_2_840_10008_1_4_1_10, [1313](#)
uid_1_2_840_10008_1_4_1_11, [1313](#)
uid_1_2_840_10008_1_4_1_12, [1313](#)
uid_1_2_840_10008_1_4_1_13, [1313](#)
uid_1_2_840_10008_1_4_1_14, [1313](#)
uid_1_2_840_10008_1_4_1_15, [1313](#)
uid_1_2_840_10008_1_4_1_16, [1313](#)
uid_1_2_840_10008_1_4_1_17, [1313](#)
uid_1_2_840_10008_1_4_1_18, [1313](#)
uid_1_2_840_10008_1_4_1_2, [1313](#)
uid_1_2_840_10008_1_4_1_3, [1313](#)
uid_1_2_840_10008_1_4_1_4, [1313](#)
uid_1_2_840_10008_1_4_1_5, [1313](#)
uid_1_2_840_10008_1_4_1_6, [1313](#)
uid_1_2_840_10008_1_4_1_7, [1313](#)
uid_1_2_840_10008_1_4_1_8, [1313](#)
uid_1_2_840_10008_1_4_1_9, [1313](#)
uid_1_2_840_10008_1_4_2_1, [1313](#)
uid_1_2_840_10008_1_4_2_2, [1313](#)
uid_1_2_840_10008_1_5_1, [1323](#)
uid_1_2_840_10008_1_5_2, [1322](#)
uid_1_2_840_10008_1_5_3, [1322](#)
uid_1_2_840_10008_1_5_4, [1322](#)
uid_1_2_840_10008_1_5_5, [1322](#)
uid_1_2_840_10008_1_5_6, [1322](#)
uid_1_2_840_10008_1_5_7, [1322](#)
uid_1_2_840_10008_1_5_8, [1322](#)

uid_1_2_840_10008_1_9, [1313](#)
 uid_1_2_840_10008_2_16_10, [1322](#)
 uid_1_2_840_10008_2_16_11, [1323](#)
 uid_1_2_840_10008_2_16_12, [1323](#)
 uid_1_2_840_10008_2_16_13, [1323](#)
 uid_1_2_840_10008_2_16_14, [1323](#)
 uid_1_2_840_10008_2_16_4, [1314](#)
 uid_1_2_840_10008_2_16_5, [1322](#)
 uid_1_2_840_10008_2_16_6, [1322](#)
 uid_1_2_840_10008_2_16_7, [1322](#)
 uid_1_2_840_10008_2_16_8, [1322](#)
 uid_1_2_840_10008_2_16_9, [1322](#)
 uid_1_2_840_10008_2_6_1, [1314](#)
 uid_1_2_840_10008_3_1_1_1, [1314](#)
 uid_1_2_840_10008_3_1_2_1_1, [1314](#)
 uid_1_2_840_10008_3_1_2_1_4, [1314](#)
 uid_1_2_840_10008_3_1_2_2_1, [1314](#)
 uid_1_2_840_10008_3_1_2_3_1, [1314](#)
 uid_1_2_840_10008_3_1_2_3_2, [1314](#)
 uid_1_2_840_10008_3_1_2_3_3, [1314](#)
 uid_1_2_840_10008_3_1_2_3_4, [1314](#)
 uid_1_2_840_10008_3_1_2_3_5, [1314](#)
 uid_1_2_840_10008_3_1_2_5_1, [1314](#)
 uid_1_2_840_10008_3_1_2_5_4, [1314](#)
 uid_1_2_840_10008_3_1_2_5_5, [1314](#)
 uid_1_2_840_10008_3_1_2_6_1, [1314](#)
 uid_1_2_840_10008_4_2, [1314](#)
 uid_1_2_840_10008_5_1_1_1, [1314](#)
 uid_1_2_840_10008_5_1_1_14, [1315](#)
 uid_1_2_840_10008_5_1_1_15, [1315](#)
 uid_1_2_840_10008_5_1_1_16, [1315](#)
 uid_1_2_840_10008_5_1_1_16_376, [1315](#)
 uid_1_2_840_10008_5_1_1_17, [1315](#)
 uid_1_2_840_10008_5_1_1_17_376, [1315](#)
 uid_1_2_840_10008_5_1_1_18, [1315](#)
 uid_1_2_840_10008_5_1_1_18_1, [1315](#)
 uid_1_2_840_10008_5_1_1_2, [1314](#)
 uid_1_2_840_10008_5_1_1_22, [1315](#)
 uid_1_2_840_10008_5_1_1_23, [1315](#)
 uid_1_2_840_10008_5_1_1_24, [1315](#)
 uid_1_2_840_10008_5_1_1_24_1, [1315](#)
 uid_1_2_840_10008_5_1_1_25, [1315](#)
 uid_1_2_840_10008_5_1_1_26, [1315](#)
 uid_1_2_840_10008_5_1_1_27, [1315](#)
 uid_1_2_840_10008_5_1_1_29, [1315](#)
 uid_1_2_840_10008_5_1_1_30, [1315](#)
 uid_1_2_840_10008_5_1_1_31, [1315](#)
 uid_1_2_840_10008_5_1_1_32, [1315](#)
 uid_1_2_840_10008_5_1_1_33, [1315](#)
 uid_1_2_840_10008_5_1_1_4, [1314](#)
 uid_1_2_840_10008_5_1_1_40, [1323](#)
 uid_1_2_840_10008_5_1_1_40_1, [1323](#)
 uid_1_2_840_10008_5_1_1_4_1, [1315](#)
 uid_1_2_840_10008_5_1_1_4_2, [1315](#)
 uid_1_2_840_10008_5_1_1_9, [1315](#)
 uid_1_2_840_10008_5_1_1_9_1, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_10, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_104_1, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_104_2, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_104_3, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_11, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_1, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_10, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_11, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_2, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_3, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_4, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_5, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_6, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_7, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_8, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_9, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_128, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_128_1, [1322](#)
 uid_1_2_840_10008_5_1_4_1_1_129, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1_1, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2_1, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_3, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_77, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_130, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_131, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_1, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_2,

- [1317](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_3,
[1322](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_4,
[1323](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_5,
[1323](#)
uid_1_2_840_10008_5_1_4_1_1_14_1,
[1323](#)
uid_1_2_840_10008_5_1_4_1_1_14_2,
[1323](#)
uid_1_2_840_10008_5_1_4_1_1_1_1, [1315](#)
uid_1_2_840_10008_5_1_4_1_1_1_1_1,
[1315](#)
uid_1_2_840_10008_5_1_4_1_1_1_2, [1315](#)
uid_1_2_840_10008_5_1_4_1_1_1_2_1,
[1315](#)
uid_1_2_840_10008_5_1_4_1_1_1_3, [1315](#)
uid_1_2_840_10008_5_1_4_1_1_1_3_1,
[1316](#)
uid_1_2_840_10008_5_1_4_1_1_2, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_20, [1317](#)
uid_1_2_840_10008_5_1_4_1_1_200_1,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_200_2,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_200_3,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_200_4,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_200_5,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_200_6,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_2_1, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_2_2, [1322](#)
uid_1_2_840_10008_5_1_4_1_1_3, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_30, [1323](#)
uid_1_2_840_10008_5_1_4_1_1_3_1, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_4, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_40, [1323](#)
uid_1_2_840_10008_5_1_4_1_1_481_1,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_10,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_481_11,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_481_2,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_3,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_4,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_5,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_6,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_7,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_8,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_481_9,
[1319](#)
uid_1_2_840_10008_5_1_4_1_1_4_1, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_4_2, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_4_3, [1327](#)
uid_1_2_840_10008_5_1_4_1_1_4_4, [1322](#)
uid_1_2_840_10008_5_1_4_1_1_5, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_501_1,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_2_1,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_2_2,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_3,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_4,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_5,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_501_6,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_6, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_601_1,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_601_2,
[1325](#)
uid_1_2_840_10008_5_1_4_1_1_66, [1317](#)
uid_1_2_840_10008_5_1_4_1_1_66_1,
[1317](#)
uid_1_2_840_10008_5_1_4_1_1_66_2,
[1317](#)
uid_1_2_840_10008_5_1_4_1_1_66_3,
[1317](#)
uid_1_2_840_10008_5_1_4_1_1_66_4,
[1317](#)
uid_1_2_840_10008_5_1_4_1_1_66_5,
[1322](#)
uid_1_2_840_10008_5_1_4_1_1_66_6,
[1324](#)
uid_1_2_840_10008_5_1_4_1_1_67, [1317](#)
uid_1_2_840_10008_5_1_4_1_1_68_1,
[1324](#)
uid_1_2_840_10008_5_1_4_1_1_68_2,
[1324](#)
uid_1_2_840_10008_5_1_4_1_1_6_1, [1316](#)
uid_1_2_840_10008_5_1_4_1_1_6_2, [1322](#)
uid_1_2_840_10008_5_1_4_1_1_7, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_77_1,
[1317](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_2,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_3,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_4,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_5,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_6,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_7,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_8,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_6,
[1322](#)
 uid_1_2_840_10008_5_1_4_1_1_77_2,
[1317](#)
 uid_1_2_840_10008_5_1_4_1_1_78_1,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_2,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_3,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_4,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_5,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_6,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_7,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_8,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_79_1,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_7_1, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_2, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_3, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_4, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_8, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_80_1,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_81_1,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_82_1,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_1,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_11,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_2,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_22,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_3,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_33,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_34,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_35,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_4,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_40,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_50,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_59,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_65,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_67,
[1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_68,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_69,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_70,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_71,
[1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_72,
[1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_73,
[1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_74,
[1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_75,

- [1325](#)
- [uid_1_2_840_10008_5_1_4_1_1_9](#), [1316](#)
- [uid_1_2_840_10008_5_1_4_1_1_90_1](#),
- [1325](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_1](#), [1316](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_1_1](#),
- [1316](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_1_2](#),
- [1316](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_1_3](#),
- [1316](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_2_1](#),
- [1317](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_3_1](#),
- [1317](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_4_1](#),
- [1317](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_4_2](#),
- [1323](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_5_1](#),
- [1323](#)
- [uid_1_2_840_10008_5_1_4_1_1_9_6_1](#),
- [1323](#)
- [uid_1_2_840_10008_5_1_4_1_2_1_1](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_1_2](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_1_3](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_2_1](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_2_2](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_2_3](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_3_1](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_3_2](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_3_3](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_1_2_4_2](#), [1325](#)
- [uid_1_2_840_10008_5_1_4_1_2_4_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_1_2_5_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_20_1](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_20_2](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_20_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_31](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_32](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_32_1](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_32_2](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_32_3](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_33](#), [1319](#)
- [uid_1_2_840_10008_5_1_4_34_1](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_10](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_2](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_3](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_4](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_4_1](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_4_2](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_4_3](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_4_4](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_5](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_34_5_1](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_6](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_6_1](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_6_2](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_6_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_6_4](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_7](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_8](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_34_9](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_37_1](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_37_2](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_37_3](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_38_1](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_38_2](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_38_3](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_38_4](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_39_1](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_39_2](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_39_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_39_4](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_41](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_42](#), [1320](#)
- [uid_1_2_840_10008_5_1_4_43_1](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_43_2](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_43_3](#), [1326](#)
- [uid_1_2_840_10008_5_1_4_43_4](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_44_1](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_44_2](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_44_3](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_44_4](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_45_1](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_45_2](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_45_3](#), [1327](#)
- [uid_1_2_840_10008_5_1_4_45_4](#), [1327](#)
- [uid_1_2_840_10008_7_1_1](#), [1327](#)
- [uid_1_2_840_10008_7_1_2](#), [1327](#)
- [uid_1_2_840_10008_8_1_1](#), [1327](#)
- UIDs, [1327](#)
- UltrasoundImageStorage, [1304](#)
- UltrasoundImageStorageRetired, [1304](#)
- UltrasoundMultiframeImageStorage, [1304](#)
- UltrasoundMultiframeImageStorageRetired,
- [1304](#)
- UnifiedProcedureStepEventSOPClass, [1306](#)
- UnifiedProcedureStepEventSOPClass1, [1311](#)
- UnifiedProcedureStepPullSOPClass, [1306](#)
- UnifiedProcedureStepPullSOPClass1, [1311](#)
- UnifiedProcedureStepPushSOPClass, [1306](#)
- UnifiedProcedureStepPushSOPClass1, [1311](#)
- UnifiedProcedureStepWatchSOPClass, [1306](#)
- UnifiedProcedureStepWatchSOPClass1, [1311](#)
- UnifiedWorklistandProcedureStepServiceClass,
- [1306](#)

- UnifiedWorklistandProcedureStepServiceClass1, 1311
- UnifiedWorklistandProcedureStepSOPInstance, 1306
- UniversalCoordinatedTime, 1311
- UPSFilteredGlobalSubscriptionSOPInstance, 1311
- VerificationSOPClass, 1301
- VideoEndoscopicImageStorage, 1305
- VideoMicroscopicImageStorage, 1305
- VideoPhotographicImageStorage, 1305
- VisualAcuityMeasurementsStorage, 1309
- VLEndoscopicImageStorage, 1305
- VLIImageStorageTrialRetired, 1305
- VLMicroscopicImageStorage, 1305
- VLMultiframeImageStorageTrialRetired, 1305
- VLPhotographicImageStorage, 1305
- VLSlideCoordinatesMicroscopicImageStorage, 1305
- VLWholeSlideMicroscopyImageStorage, 1308
- VOILUTBoxSOPClass, 1303
- VolumeRenderingVolumetricPresentation-StateStorage, 1309
- WaveformStorageTrialRetired, 1304
- WideFieldOphthalmicPhotography3DCoordinatesImageStorage, 1309
- WideFieldOphthalmicPhotographyStereographicProjectionImageStorage, 1309
- WinterColorPaletteSOPInstance, 1308
- XAXRFGrayscaleSoftcopyPresentationStateStorage, 1309
- XMLEncoding, 1302
- XRay3DAngiographicImageStorage, 1305
- XRay3DCraniofacialImageStorage, 1305
- XRayAngiographicBiPlaneImageStorageRetired, 1305
- XRayAngiographicImageStorage, 1305
- XRayRadiationDoseSRStorage, 1306
- XRayRadiofluoroscopicImageStorage, 1305
- gdcmm::UNExplicitDataElement, 1395
 - GetLength, 1398
 - Read, 1398
 - ReadPreValue, 1398
 - ReadValue, 1398
 - ReadWithLength, 1398
- gdcmm::UNExplicitImplicitDataElement, 1399
 - GetLength, 1402
 - Read, 1402
 - ReadPreValue, 1402
 - ReadValue, 1402
- gdcmm::Unpacker12Bits, 1403
 - Pack, 1403
 - Unpack, 1403
- gdcmm::Usage, 1404
 - Conditional, 1405
 - GetUsageString, 1406
 - GetUsageType, 1406
 - Invalid, 1405
 - Mandatory, 1405
 - operator UsageType, 1406
 - operator<<, 1406
 - Usage, 1405
 - UsageType, 1405
 - UserOption, 1405
- gdcmm::UserEvent, 1407
- gdcmm::UUIDGenerator, 1410
 - Generate, 1411
 - IsValid, 1411
- gdcmm::Validate, 1411
 - ~Validate, 1412
 - F, 1413
 - GetValidatedFile, 1413
 - SetFile, 1413
 - V, 1413
 - Validate, 1412
 - Validation, 1413
- gdcmm::Value, 1414
 - ~Value, 1415
 - Clear, 1416
 - DataElement, 1417
 - GetLength, 1416
 - operator==, 1416
 - SetLength, 1416
 - SetLengthOnly, 1416
 - Value, 1415
- gdcmm::ValueIO< TDE, TSwap, TType >, 1417
 - Read, 1417
 - Write, 1417
- gdcmm::Version, 1419
 - ~Version, 1419
 - GetBuildVersion, 1419
 - GetMajorVersion, 1419
 - GetMinorVersion, 1420
 - GetVersion, 1420
 - operator<<, 1420
 - Print, 1420
 - Version, 1419
- gdcmm::VL, 1420
 - GetLength, 1422
 - GetVL16Max, 1422
 - GetVL32Max, 1422
 - IsOdd, 1422
 - IsUndefined, 1422
 - operator uint32_t, 1423
 - operator<<, 1424
 - operator++, 1423
 - operator+==, 1423
 - Read, 1423

- Read16, [1423](#)
- SetToUndefined, [1423](#)
- Type, [1421](#)
- VL, [1422](#)
- Write, [1424](#)
- Write16, [1424](#)
- gdcmm::VM, [1424](#)
 - Compatible, [1428](#)
 - GetIndex, [1428](#)
 - GetLength, [1428](#)
 - GetNumberOfElementsFromArray, [1428](#)
 - GetVMString, [1428](#)
 - GetVMType, [1429](#)
 - GetVMTypeFromLength, [1429](#)
 - IsValid, [1429](#)
 - operator VMType, [1429](#)
 - operator<<, [1429](#)
 - VM, [1428](#)
 - VM0, [1426](#)
 - VM1, [1426](#)
 - VM10, [1427](#)
 - VM12, [1427](#)
 - VM16, [1427](#)
 - VM18, [1427](#)
 - VM1_2, [1427](#)
 - VM1_3, [1427](#)
 - VM1_32, [1427](#)
 - VM1_4, [1427](#)
 - VM1_5, [1427](#)
 - VM1_8, [1427](#)
 - VM1_99, [1427](#)
 - VM1_n, [1427](#)
 - VM2, [1426](#)
 - VM24, [1427](#)
 - VM256, [1427](#)
 - VM28, [1427](#)
 - VM2_2n, [1427](#)
 - VM2_n, [1427](#)
 - VM3, [1426](#)
 - VM30_30n, [1427](#)
 - VM32, [1427](#)
 - VM35, [1427](#)
 - VM3_3n, [1427](#)
 - VM3_4, [1427](#)
 - VM3_n, [1427](#)
 - VM4, [1426](#)
 - VM47_47n, [1427](#)
 - VM4_4n, [1427](#)
 - VM5, [1426](#)
 - VM6, [1427](#)
 - VM6_6n, [1427](#)
 - VM6_n, [1427](#)
 - VM7_7n, [1427](#)
 - VM8, [1427](#)
 - VM9, [1427](#)
 - VM99, [1427](#)
 - VM_END, [1428](#)
 - VMType, [1426](#)
- gdcmm::VMToLength< T >, [1430](#)
- gdcmm::VR, [1430](#)
 - AE, [1432](#)
 - AS, [1432](#)
 - AT, [1432](#)
 - CanDisplay, [1434](#)
 - Compatible, [1434](#)
 - CS, [1432](#)
 - DA, [1432](#)
 - DS, [1432](#)
 - DT, [1432](#)
 - FD, [1432](#)
 - FL, [1432](#)
 - GetLength, [1434](#)
 - GetSize, [1434](#)
 - GetSizeof, [1434](#)
 - GetVRString, [1435](#)
 - GetVRStringFromFile, [1435](#)
 - GetVRType, [1435](#)
 - GetVRTypeFromFile, [1435](#)
 - INVALID, [1432](#)
 - IS, [1432](#)
 - IsASCII, [1435](#)
 - IsASCII2, [1435](#)
 - IsBinary, [1435](#)
 - IsBinary2, [1435](#)
 - IsDual, [1436](#)
 - IsSwap, [1436](#)
 - IsValid, [1436](#)
 - IsVRFile, [1436](#)
 - LO, [1432](#)
 - LT, [1432](#)
 - OB, [1432](#)
 - OB_OW, [1433](#)
 - OD, [1432](#)
 - OF, [1432](#)
 - OL, [1433](#)
 - operator VRType, [1436](#)
 - operator<<, [1437](#)
 - OV, [1433](#)
 - OW, [1433](#)
 - PN, [1433](#)
 - Read, [1436](#)
 - SH, [1433](#)
 - SL, [1433](#)
 - SQ, [1433](#)
 - SS, [1433](#)
 - ST, [1433](#)
 - SV, [1433](#)
 - TM, [1433](#)

- UC, [1433](#)
- UI, [1433](#)
- UL, [1433](#)
- UN, [1433](#)
- UR, [1433](#)
- US, [1433](#)
- US_OW, [1433](#)
- US_SS, [1433](#)
- US_SS_OW, [1433](#)
- UT, [1433](#)
- UV, [1433](#)
- VL16, [1433](#)
- VL32, [1433](#)
- VR, [1433](#)
- VR_END, [1433](#)
- VR_VM1, [1433](#)
- VRALL, [1433](#)
- VRASCII, [1433](#)
- VRBINARY, [1433](#)
- VRType, [1432](#)
- Write, [1436](#)
- gdcm::VR16ExplicitDataElement, [1437](#)
 - GetLength, [1440](#)
 - Read, [1440](#)
 - ReadPreValue, [1440](#)
 - ReadValue, [1440](#)
 - ReadWithLength, [1440](#)
- gdcm::VRToEncoding< T >, [1441](#)
- gdcm::VRToType< T >, [1441](#)
- gdcm::VRVLSIZE< 0 >, [1442](#)
 - Read, [1443](#)
 - Write, [1443](#)
- gdcm::VRVLSIZE< 1 >, [1444](#)
 - Read, [1444](#)
 - Write, [1444](#)
- gdcm::VRVLSIZE< T >, [1442](#)
- gdcm::Waveform, [1553](#)
 - Waveform, [1554](#)
- gdcm::WLMFindQuery, [1554](#)
 - GetAbstractSyntaxUID, [1557](#)
 - GetTagListByLevel, [1557](#)
 - GetValidDataSet, [1557](#)
 - InitializeDataSet, [1557](#)
 - QueryFactory, [1558](#)
 - ValidateQuery, [1557](#)
 - WLMFindQuery, [1557](#)
- gdcm::Writer, [1558](#)
 - ~Writer, [1561](#)
 - CheckFileMetaInformationOff, [1561](#)
 - CheckFileMetaInformationOn, [1561](#)
 - GetCheckFileMetaInformation, [1561](#)
 - GetFile, [1561](#)
 - GetStreamPtr, [1562](#)
 - Ofstream, [1564](#)
 - SetCheckFileMetaInformation, [1562](#)
 - SetFile, [1562](#)
 - SetFileName, [1562](#)
 - SetStream, [1563](#)
 - SetWriteDataSetOnly, [1563](#)
 - Stream, [1564](#)
 - StreamImageWriter, [1564](#)
 - Write, [1563](#)
 - Writer, [1561](#)
- gdcm::XMLDictReader, [1564](#)
 - ~XMLDictReader, [1566](#)
 - CharacterDataHandler, [1566](#)
 - EndElement, [1566](#)
 - GetDict, [1566](#)
 - HandleDescription, [1566](#)
 - HandleEntry, [1566](#)
 - StartElement, [1567](#)
 - XMLDictReader, [1566](#)
- gdcm::XMLPrinter, [1567](#)
 - ~XMLPrinter, [1568](#)
 - F, [1570](#)
 - GetPrintStyle, [1569](#)
 - HandleBulkData, [1569](#)
 - LOADBULKDATA, [1568](#)
 - OnlyUUID, [1568](#)
 - Print, [1569](#)
 - PrintDataElement, [1569](#)
 - PrintDataSet, [1569](#)
 - PrintSQ, [1569](#)
 - PrintStyle, [1570](#)
 - PrintStyles, [1568](#)
 - SetFile, [1569](#)
 - SetStyle, [1570](#)
 - XMLPrinter, [1568](#)
- gdcm::XMLPrivateDictReader, [1571](#)
 - ~XMLPrivateDictReader, [1572](#)
 - CharacterDataHandler, [1573](#)
 - EndElement, [1573](#)
 - GetPrivateDict, [1573](#)
 - HandleDescription, [1573](#)
 - HandleEntry, [1573](#)
 - StartElement, [1573](#)
 - XMLPrivateDictReader, [1572](#)
- gdcm_assert
 - gdcmException.h, [1602](#)
- gdcm_debug_assert
 - gdcmException.h, [1603](#)
- GDCM_DIFFERENT
 - gdcm, [89](#)
- GDCM_DO_JOIN
 - gdcmStaticAssert.h, [1631](#)
- GDCM_DO_JOIN2
 - gdcmStaticAssert.h, [1631](#)
- GDCM_EQUAL

- gdcM, 89
- GDCM_EXPORT
 - gdcMWin32.h, 1659
- gdcM_forced_assert
 - gdcMException.h, 1603
- GDCM_FUNCTION
 - gdcMTrace.h, 1650
- GDCM_GREATER
 - gdcM, 89
- GDCM_GREATEROREQUAL
 - gdcM, 89
- GDCM_JOIN
 - gdcMStaticAssert.h, 1631
- GDCM_LEGACY
 - gdcMLegacyMacro.h, 1610
- GDCM_LEGACY_BODY
 - gdcMLegacyMacro.h, 1610
- GDCM_LEGACY_REPLACED_BODY
 - gdcMLegacyMacro.h, 1610
- GDCM_LESS
 - gdcM, 89
- GDCM_LESSOREQUAL
 - gdcM, 89
- GDCM_NOOP_STATEMENT
 - gdcMLegacyMacro.h, 1611
- GDCM_STATIC_ASSERT
 - gdcM::Attribute< Group, Element, TVR, TVM
>, 161
 - gdcM::Attribute< Group, Element, TVR,
VM::VM1 >, 171
 - gdcM::Attribute< Group, Element, TVR,
VM::VM1_3 >, 178
 - gdcM::Attribute< Group, Element, TVR,
VM::VM1_8 >, 184
 - gdcM::Attribute< Group, Element, TVR,
VM::VM1_n >, 189
 - gdcM::Attribute< Group, Element, TVR,
VM::VM2_2n >, 198
 - gdcM::Attribute< Group, Element, TVR,
VM::VM2_n >, 204
 - gdcM::Attribute< Group, Element, TVR,
VM::VM3_3n >, 210
 - gdcM::Attribute< Group, Element, TVR,
VM::VM3_n >, 216
 - gdcMStaticAssert.h, 1631
- gdcMAAbortPDU.h, 2086, 2087
- gdcMAAssociateACPDU.h, 2087, 2088
- gdcMAAssociateRJPDU.h, 2090
- gdcMAAssociateRQPDU.h, 2091, 2092
- gdcMAbstractSyntax.h, 2094, 2095
- gdcMAnonymizeEvent.h, 1889, 1891
- gdcMAnonymizer.h, 1891, 1892
- gdcMApplicationContext.h, 2096, 2097
- gdcMApplicationEntity.h, 1894
- gdcMReleaseRPPDU.h, 2097, 2098
- gdcMReleaseRQPDU.h, 2099, 2100
- gdcMARTIMTimer.h, 2101
- gdcMASN1.h, 1575, 1576
- gdcMAssertAlwaysMacro
 - gdcMTrace.h, 1650
- gdcMAssertMacro
 - gdcMTrace.h, 1650
- gdcMASynchronousOperationsWindowSub.h, 2102,
2103
- gdcMAttribute.h, 1697, 1699
- gdcMAudioCodec.h, 1895, 1896
- gdcMBase64.h, 1577
- gdcMBaseCompositeMessage.h, 2103, 2104
- gdcMBaseNormalizedMessage.h, 2105, 2106
- gdcMBasePDU.h, 2107, 2108
- gdcMBaseQuery.h, 2108, 2110
- gdcMBaseRootQuery.h, 2111, 2112
- gdcMBasicOffsetTable.h, 1712, 1713
- gdcMBitmap.h, 1896, 1897
- gdcMBitmapToBitmapFilter.h, 1900
- gdcMBoxRegion.h, 1578, 1579
- gdcMByteBuffer.h, 1714, 1715
- gdcMByteSwap.h, 1579, 1580
- gdcMByteSwapFilter.h, 1717, 1718
- gdcMByteValue.h, 1718, 1719
- gdcMCAPICryptoFactory.h, 1581, 1582
- gdcMCAPICryptographicMessageSyntax.h, 1582,
1583
- gdcMCEchoMessages.h, 2113, 2114
- gdcMCFindMessages.h, 2114, 2115
- gdcMCleaner.h, 1901, 1902
- gdcMCMoveMessages.h, 2116, 2117
- gdcMCodec.h, 1903, 1904
- gdcMCoder.h, 1904, 1905
- gdcMCodeString.h, 1723
- gdcMCommand.h, 1585, 1586
- gdcMCommandDataSet.h, 2118
- gdcMCompositeMessageFactory.h, 2119, 2120
- gdcMCompositeNetworkFunctions.h, 2121, 2122
- gdcMConstCharWrapper.h, 1906
- gdcMCP246ExplicitDataElement.h, 1725
- gdcMCryptoFactory.h, 1588, 1589
- gdcMCryptographicMessageSyntax.h, 1590, 1591
- gdcMCSAElement.h, 1726, 1727
- gdcMCSAHeader.h, 1729, 1730
- gdcMCSAHeaderDict.h, 1660, 1662
- gdcMCSAHeaderDictEntry.h, 1663, 1664
- gdcMStoreMessages.h, 2123
- gdcMCurve.h, 1907, 1908
- gdcMDataElement.h, 1732, 1733
- gdcMDataEvent.h, 1592, 1593
- gdcMDataSet.h, 1735, 1737
- gdcMDataSetEvent.h, 1740, 1741

- gdcmDataSetHelper.h, 1910
- gdcmDebugMacro
 - gdcmTrace.h, 1651
- gdcmDecoder.h, 1911, 1912
- gdcmDefinedTerms.h, 1838, 1839
- gdcmDeflateStream.h, 1594
- gdcmDefs.h, 1839, 1841
- gdcmDeltaEncodingCodec.h, 1913
- gdcmDICOMDIR.h, 1914, 1915
- gdcmDICOMDIRGenerator.h, 1915, 1916
- gdcmDict.h, 1666, 1667
- gdcmDictConverter.h, 1671, 1672
- gdcmDictEntry.h, 1673, 1674
- gdcmDictPrinter.h, 1917, 1918
- gdcmDicts.h, 1676, 1677
- gdcmDIMSE.h, 2124, 2125
- gdcmDirectionCosines.h, 1918, 1919
- gdcmDirectory.h, 1594, 1595
- gdcmDirectoryHelper.h, 1920
- gdcmDPath.h, 1921, 1923
- gdcmDummyValueGenerator.h, 1597
- gdcmDumper.h, 1924
- gdcmElement.h, 1742, 1743
- gdcmEmptyMaskGenerator.h, 1925, 1926
- gdcmEncapsulatedDocument.h, 1927
- gdcmEnumeratedValues.h, 1842
- gdcmEquipmentManufacturer.h, 1928
- gdcmErrorMacro
 - gdcmTrace.h, 1651
- gdcmEvent.h, 1598, 1600
 - gdcmEventMacro, 1599
- gdcmEventMacro
 - gdcmEvent.h, 1599
- gdcmException.h, 1601, 1603
 - gdcm_assert, 1602
 - gdcm_debug_assert, 1603
 - gdcm_forced_assert, 1603
- gdcmExplicitDataElement.h, 1754, 1755
- gdcmExplicitImplicitDataElement.h, 1756, 1757
- gdcmFiducials.h, 1929, 1930
- gdcmFile.h, 1758, 1759
- gdcmFileAnonymizer.h, 1930, 1931
- gdcmFileChangeTransferSyntax.h, 1932, 1933
- gdcmFileDecompressLookupTable.h, 1934, 1935
- gdcmFileDerivation.h, 1936
- gdcmFileExplicitFilter.h, 1938
- gdcmFileMetaInformation.h, 1759, 1761
- gdcmFilename.h, 1605
- gdcmFileNameEvent.h, 1606, 1607
- gdcmFilenameGenerator.h, 1608, 1609
- gdcmFileSet.h, 1762, 1764
- gdcmFileStreamer.h, 1939, 1940
- gdcmFindPatientRootQuery.h, 2127, 2128
- gdcmFindStudyRootQuery.h, 2128, 2129
- gdcmFragment.h, 1764, 1766
- gdcmGlobal.h, 1679, 1680
- gdcmGroupDict.h, 1681, 1682
- gdcmIconImage.h, 1941, 1942
- gdcmIconImageFilter.h, 1943, 1944
- gdcmIconImageGenerator.h, 1945, 1946
- gdcmImage.h, 1946, 1948
- gdcmImageApplyLookupTable.h, 1949
- gdcmImageChangePhotometricInterpretation.h, 1950, 1951
- gdcmImageChangePlanarConfiguration.h, 1953
- gdcmImageChangeTransferSyntax.h, 1954, 1955
- gdcmImageCodec.h, 1956, 1957
- gdcmImageConverter.h, 1959, 1960
- gdcmImageFragmentSplitter.h, 1961
- gdcmImageHelper.h, 1962, 1963
- gdcmImageReader.h, 1964, 1965
- gdcmImageRegionReader.h, 1966, 1967
- gdcmImageToImageFilter.h, 1968, 1969
- gdcmImageWriter.h, 1969, 1970
- gdcmImplementationClassUIDSub.h, 2130, 2131
- gdcmImplementationUIDSub.h, 2132
- gdcmImplementationVersionNameSub.h, 2133, 2134
- gdcmImplicitDataElement.h, 1769
- gdcmIOD.h, 1843, 1844
- gdcmIODEntry.h, 1846, 1848
- gdcmIODs.h, 1848, 1850
- gdcmIPPSorter.h, 1971, 1972
- gdcmItem.h, 1770, 1771
- gdcmJPEG12Codec.h, 1973
- gdcmJPEG16Codec.h, 1974, 1975
- gdcmJPEG2000Codec.h, 1976
- gdcmJPEG8Codec.h, 1978
- gdcmJPEGCodec.h, 1979, 1980
- gdcmJPEGLSCodec.h, 1982
- gdcmJSON.h, 1983, 1984
- gdcmKAKADUCodec.h, 1985, 1986
- gdcmLegacyMacro.h, 1609, 1611
 - GDCM_LEGACY, 1610
 - GDCM_LEGACY_BODY, 1610
 - GDCM_LEGACY_REPLACED_BODY, 1610
 - GDCM_NOOP_STATEMENT, 1611
- gdcmLO.h, 1776
- gdcmLookupTable.h, 1986, 1987
- gdcmMacro.h, 1851, 1852
- gdcmMacroEntry.h, 1854, 1856
 - GDCMMACROENTRY_H, 1855
- GDCMMACROENTRY_H
 - gdcmMacroEntry.h, 1855
- gdcmMacros.h, 1857, 1858
- gdcmMaximumLengthSub.h, 2135, 2136
- gdcmMD5.h, 1612, 1613
- gdcmMEC_MR3.h, 1989
- gdcmMediaStorage.h, 1777, 1778

- gdcmMeshPrimitive.h, 1990, 1991
- gdcmModalityPerformedProcedureStepCreate-
Query.h, 2137, 2138
- gdcmModalityPerformedProcedureStepSetQuery.h,
2138, 2139
- gdcmModule.h, 1859, 1861
- gdcmModuleEntry.h, 1862, 1864
- gdcmModules.h, 1865, 1866
- gdcmMovePatientRootQuery.h, 2140
- gdcmMoveStudyRootQuery.h, 2141, 2142
- gdcmMrProtocol.h, 1781, 1782
- gdcmNActionMessages.h, 2142, 2143
- gdcmNCreateMessages.h, 2144
- gdcmNDeleteMessages.h, 2145, 2146
- gdcmNestedModuleEntries.h, 1867, 1869
- gdcmNetworkEvents.h, 2146, 2147
- gdcmNetworkStateID.h, 2148, 2149
- gdcmNEventReportMessages.h, 2150, 2151
- gdcmNGetMessages.h, 2152
- gdcmNormalizedMessageFactory.h, 2153, 2154
- gdcmNormalizedNetworkFunctions.h, 2154, 2155
- gdcmNSetMessages.h, 2156, 2157
- gdcmObject.h, 1613, 1614
- gdcmOpenSSLCryptoFactory.h, 1616, 1617
- gdcmOpenSSLCryptographicMessageSyntax.h, 1617,
1619
- gdcmOpenSSLTP7CryptoFactory.h, 1619, 1620
- gdcmOpenSSLTP7CryptographicMessageSyntax.h,
1621, 1622
- gdcmOrientation.h, 1993
- gdcmOverlay.h, 1994, 1995
- gdcmParseException.h, 1783, 1784
- gdcmParser.h, 1785, 1786
- gdcmPatient.h, 1869, 1870
- gdcmPDataTFPDU.h, 2157, 2158
- gdcmPDBelement.h, 1788, 1789
- gdcmPDBHeader.h, 1790, 1791
- gdcmPDFCodec.h, 1997
- gdcmPDUFactory.h, 2159, 2160
- gdcmPersonName.h, 1998, 1999
- gdcmPGXCodec.h, 2000, 2001
- gdcmPhotometricInterpretation.h, 2001, 2003
- gdcmPixelFormat.h, 2004, 2005
- gdcmPixmap.h, 2008, 2009
- gdcmPixmapReader.h, 2010, 2011
- gdcmPixmapToPixmapFilter.h, 2012, 2013
- gdcmPixmapWriter.h, 2013, 2014
- gdcmPNMCodec.h, 2016
- gdcmPreamble.h, 1792, 1793
- gdcmPresentationContext.h, 2161, 2162
- gdcmPresentationContextAC.h, 2163, 2164
- gdcmPresentationContextGenerator.h, 2165, 2166
- gdcmPresentationContextRQ.h, 2166, 2167
- gdcmPresentationDataValue.h, 2169, 2170
- gdcmPrinter.h, 2017, 2018
- gdcmPrivateTag.h, 1794, 1795
- gdcmProgressEvent.h, 1623, 1624
- gdcmPVRGCodec.h, 2020
- gdcmPythonFilter.h, 2266, 2267
- gdcmQueryBase.h, 2171, 2172
- gdcmQueryFactory.h, 2173, 2174
- gdcmQueryImage.h, 2175, 2176
- gdcmQueryPatient.h, 2177, 2178
- gdcmQuerySeries.h, 2179, 2180
- gdcmQueryStudy.h, 2181, 2182
- gdcmRAWCodec.h, 2021, 2022
- gdcmReader.h, 1796, 1797
- gdcmRegion.h, 1624, 1626
- gdcmRescaler.h, 2022, 2023
- gdcmRLECodec.h, 2025
- gdcmRoleSelectionSub.h, 2182, 2183
- gdcmScanner.h, 2026, 2027
- gdcmScanner2.h, 2029, 2030
- gdcmSegment.h, 2032, 2034
- gdcmSegmentedPaletteColorLookupTable.h, 2036
- gdcmSegmentHelper.h, 2037, 2038
- gdcmSegmentReader.h, 2039, 2041
- gdcmSegmentWriter.h, 2041, 2043
- gdcmSequenceOfFragments.h, 1798, 1799
- gdcmSequenceOfItems.h, 1803, 1804
- gdcmSerieHelper.h, 2043, 2045
- gdcmSeries.h, 1871, 1872
- gdcmServiceClassApplicationInformation.h, 2184,
2185
- gdcmServiceClassUser.h, 2185, 2186
- gdcmSHA1.h, 1627
- gdcmSimpleSubjectWatcher.h, 2046, 2047
- gdcmSmartPointer.h, 1628, 1629
- gdcmSOPClassExtendedNegotiationSub.h, 2188
- gdcmSOPClassUIDToIOD.h, 1683
- gdcmSorter.h, 2048, 2050
- gdcmSpacing.h, 2051
- gdcmSpectroscopy.h, 2053
- gdcmSplitMosaicFilter.h, 2054
- gdcmStaticAssert.h, 1630, 1632
 - GDCM_DO_JOIN, 1631
 - GDCM_DO_JOIN2, 1631
 - GDCM_JOIN, 1631
 - GDCM_STATIC_ASSERT, 1631
- gdcmStreamImageReader.h, 2056
- gdcmStreamImageWriter.h, 2057, 2058
- gdcmStrictScanner.h, 2059, 2060
- gdcmStrictScanner2.h, 2062, 2063
- gdcmString.h, 1632, 1634
- gdcmStringFilter.h, 2065, 2066
- gdcmStudy.h, 1873, 1874
- gdcmSubject.h, 1636
- gdcmSurface.h, 2066, 2068

- gdcmSurfaceHelper.h, 2071
- gdcmSurfaceReader.h, 2073, 2074
- gdcmSurfaceWriter.h, 2075, 2076
- gdcmSwapCode.h, 1637, 1638
- gdcmSwapper.h, 1639, 1640
- gdcmSystem.h, 1642
- gdcmTable.h, 1874, 1875
- gdcmTableEntry.h, 1876, 1878
- gdcmTableReader.h, 1878, 1880
- gdcmTag.h, 1807, 1809
- gdcmTagPath.h, 2076, 2077
- gdcmTagToVR.h, 1812
- gdcmTerminal.h, 1644, 1645
- gdcmTestDriver.h, 1646
- gdcmTesting.h, 1647
- gdcmTrace.h, 1649, 1653
 - GDCM_FUNCTION, 1650
 - gdcmAssertAlwaysMacro, 1650
 - gdcmAssertMacro, 1650
 - gdcmDebugMacro, 1651
 - gdcmErrorMacro, 1651
 - gdcmWarningMacro, 1652
- gdcmTransferSyntax.h, 1813, 1814
- gdcmTransferSyntaxSub.h, 2189, 2191
- gdcmType.h, 1881, 1882
- gdcmTypes.h, 1655
- gdcmUIDGenerator.h, 2078, 2079
- gdcmUIDs.h, 1684, 1685
- gdcmULAction.h, 2191, 2192
- gdcmULActionAA.h, 2193, 2194
- gdcmULActionAE.h, 2195, 2196
- gdcmULActionAR.h, 2197, 2198
- gdcmULActionDT.h, 2200
- gdcmULBasicCallback.h, 2201, 2202
- gdcmULConnection.h, 2202, 2203
- gdcmULConnectionCallback.h, 2205, 2206
- gdcmULConnectionInfo.h, 2206, 2208
- gdcmULConnectionManager.h, 2208, 2209
- gdcmULEvent.h, 2211, 2212
- gdcmULTransitionTable.h, 2213, 2214
- gdcmULWritingCallback.h, 2216
- gdcmUNExplicitDataElement.h, 1815, 1816
- gdcmUNExplicitImplicitDataElement.h, 1817, 1818
- gdcmUnpacker12Bits.h, 1656, 1657
- gdcmUsage.h, 1883, 1886
- gdcmUserInformation.h, 2217, 2218
- gdcmUUIDGenerator.h, 2080
- gdcmValidate.h, 2081, 2082
- gdcmValue.h, 1818, 1819
- gdcmValueIO.h, 1820, 1821
- gdcmVersion.h, 1657, 1658
- gdcmVL.h, 1821, 1822
- gdcmVM.h, 1824, 1826
 - TYPETOLENGTH, 1825
- gdcmVR.h, 1827, 1830
 - TYPETOENCODING, 1829
 - VRTypeTemplateCase, 1829
- gdcmVR16ExplicitDataElement.h, 1834, 1835
- gdcmWarningMacro
 - gdcmTrace.h, 1652
- gdcmWaveform.h, 2082, 2083
- gdcmWin32.h, 1659
 - GDCM_EXPORT, 1659
- gdcmWLMFindQuery.h, 2219, 2220
- gdcmWriter.h, 1836, 1837
- gdcmXMLDictReader.h, 1886, 1887
- gdcmXMLPrinter.h, 2083, 2084
- gdcmXMLPrivateDictReader.h, 1888, 1889
- GEMS
 - gdcm::Dicts, 436
 - gdcm::EquipmentManufacturer, 531
- GeneralAudioWaveformStorage
 - gdcm::UIDs, 1309
- GeneralECGWaveformStorage
 - gdcm::MediaStorage, 799
 - gdcm::UIDs, 1304
- GeneralElectricMagneticResonanceImageStorage
 - gdcm::MediaStorage, 800
- GeneralPurposePerformedProcedureStepSOPClass
 - gdcm::UIDs, 1306
- GeneralPurposeScheduledProcedureStepSOPClass
 - gdcm::UIDs, 1306
- GeneralPurposeWorklistInformationModelFIND
 - gdcm::UIDs, 1306
- GeneralPurposeWorklistManagementMetaSOPClass
 - gdcm::UIDs, 1306
- GeneralRelevantPatientInformationQuery
 - gdcm::UIDs, 1307
- Generate
 - gdcm::DICOMDIRGenerator, 418
 - gdcm::DummyValueGenerator, 453
 - gdcm::FilenameGenerator, 589
 - gdcm::IconImageGenerator, 626
 - gdcm::UIDGenerator, 1283
 - gdcm::UUIDGenerator, 1411
- GenerateFromFileNames
 - gdcm::PresentationContextGenerator, 974
- GenerateFromUID
 - gdcm::PresentationContextGenerator, 974
- GenerateUUID
 - gdcm::UIDGenerator, 1283
- GenericImplantTemplateInformationModelFIND
 - gdcm::UIDs, 1311
- GenericImplantTemplateInformationModelGET
 - gdcm::UIDs, 1311
- GenericImplantTemplateInformationModelMOVE
 - gdcm::UIDs, 1311
- GenericImplantTemplateStorage

- gdcmm::UIDs, [1311](#)
- GEPrivate3DModelStorage
 - gdcmm::MediaStorage, [800](#)
- Get
 - gdcmm::ByteBuffer, [271](#)
- GetAbbreviation
 - gdcmm::GroupDict, [620](#)
- GetAbstractSyntax
 - gdcmm::network::PresentationContextRQ, [977](#)
 - gdcmm::PresentationContext, [969](#)
- GetAbstractSyntaxUID
 - gdcmm::BaseQuery, [233](#)
 - gdcmm::FindPatientRootQuery, [604](#)
 - gdcmm::FindStudyRootQuery, [608](#)
 - gdcmm::ModalityPerformedProcedureStepCreate-Query, [820](#)
 - gdcmm::ModalityPerformedProcedureStepSet-Query, [824](#)
 - gdcmm::MovePatientRootQuery, [838](#)
 - gdcmm::MoveStudyRootQuery, [842](#)
 - gdcmm::WLMFindQuery, [1557](#)
- GetAcceptedPresentationContexts
 - gdcmm::network::ULConnection, [1375](#)
- GetAcquisitionSize
 - gdcmm::SplitMosaicFilter, [1153](#)
- GetAETitle
 - gdcmm::ServiceClassUser, [1119](#)
- GetAlgorithmFamily
 - gdcmm::Surface, [1205](#)
- GetAlgorithmName
 - gdcmm::Surface, [1205](#)
- GetAlgorithmVersion
 - gdcmm::Surface, [1206](#)
- GetALGOType
 - gdcmm::Segment, [1072](#)
- GetALGOTypeString
 - gdcmm::Segment, [1073](#)
- GetAllFilenamesFromPrivateTagToValue
 - gdcmm::Scanner2, [1064](#)
 - gdcmm::StrictScanner2, [1184](#)
- GetAllFilenamesFromPublicTagToValue
 - gdcmm::Scanner2, [1064](#)
 - gdcmm::StrictScanner2, [1184](#)
- GetAllFilenamesFromTagToValue
 - gdcmm::Scanner, [1054](#)
 - gdcmm::StrictScanner, [1174](#)
- GetAllRequiredTags
 - gdcmm::QueryBase, [1006](#)
- GetAllTags
 - gdcmm::QueryBase, [1006](#)
- GetAnatomicRegion
 - gdcmm::Segment, [1073](#)
- GetAnatomicRegionModifiers
 - gdcmm::Segment, [1073](#)
- GetAsDataElement
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [162](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [171](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [178](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [198](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [210](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [216](#)
 - gdcmm::Element< TVR, TVM >, [458](#)
 - gdcmm::Element< TVR, VM::VM1_2 >, [465](#)
 - gdcmm::Element< TVR, VM::VM1_n >, [470](#)
 - gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
 - gdcmm::Element< TVR, VM::VM2_n >, [483](#)
 - gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
 - gdcmm::Element< TVR, VM::VM3_4 >, [495](#)
 - gdcmm::Element< TVR, VM::VM3_n >, [501](#)
 - gdcmm::Element< VR::AS, VM::VM5 >, [505](#)
 - gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
 - gdcmm::Element< VR::OW, VM::VM1 >, [515](#)
 - gdcmm::network::AbstractSyntax, [130](#)
 - gdcmm::PrivateTag, [993](#)
- GetAsPoints
 - gdcmm::Curve, [366](#)
- GetAsString
 - gdcmm::CodeString, [315](#)
- GetAxisOfRotation
 - gdcmm::Surface, [1206](#)
- GetBasicApplicationLevelConfidentialityProfileAttributes
 - gdcmm::Anonymizer, [140](#)
- GetBitPosition
 - gdcmm::Overlay, [893](#)
- GetBitsAllocated
 - gdcmm::Overlay, [893](#)
 - gdcmm::PixelFormat, [933](#)
- GetBitSample
 - gdcmm::LookupTable, [781](#)
- GetBitsStored
 - gdcmm::PixelFormat, [933](#)
- GetBlob
 - gdcmm::network::PresentationDataValue, [980](#)
- GetBuffer
 - gdcmm::Bitmap, [252](#)

- gdcmm::ByteValue, [279](#)
- gdcmm::Parser, [903](#)
- gdcmm::SequenceOfFragments, [1096](#)
- GetBuffer2
 - gdcmm::Bitmap, [252](#)
- GetBufferAsRGBA
 - gdcmm::LookupTable, [781](#)
- GetBufferLength
 - gdcmm::Bitmap, [252](#)
 - gdcmm::JPEGLSCodec, [765](#)
 - gdcmm::PNMCodec, [962](#)
 - gdcmm::RLECodec, [1045](#)
- GetBuildVersion
 - gdcmm::Version, [1419](#)
- GetByteValue
 - gdcmm::CSAElement, [343](#)
 - gdcmm::DataElement, [373](#)
- GetCalledAETitle
 - gdcmm::network::AAssociateRQPDU, [124](#)
 - gdcmm::network::ULConnectionInfo, [1381](#)
 - gdcmm::ServiceClassUser, [1119](#)
- GetCalledComputerName
 - gdcmm::network::ULConnectionInfo, [1381](#)
- GetCalledIPAddress
 - gdcmm::network::ULConnectionInfo, [1381](#)
- GetCalledIPPort
 - gdcmm::network::ULConnectionInfo, [1381](#)
- GetCallingAETitle
 - gdcmm::network::AAssociateRQPDU, [124](#)
 - gdcmm::network::ULConnectionInfo, [1381](#)
- GetCanonMECMR3Tag
 - gdcmm::MEC_MR3, [794](#)
- GetCenterOfRotation
 - gdcmm::Surface, [1206](#)
- GetCharacterFromCurrentLocale
 - gdcmm::QueryFactory, [1008](#)
- GetCheckFileMetaInformation
 - gdcmm::Writer, [1561](#)
- GetCipherType
 - gdcmm::CAPICryptographicMessageSyntax, [288](#)
 - gdcmm::CryptographicMessageSyntax, [339](#)
 - gdcmm::OpenSSLCryptographicMessageSyntax, [879](#)
 - gdcmm::OpenSSLTP7CryptographicMessageSyntax, [885](#)
- GetCodec
 - gdcmm::FileChangeTransferSyntax, [560](#)
- GetColorLevel
 - vtkImageColorViewer, [1514](#)
- GetColorWindow
 - vtkImageColorViewer, [1514](#)
- GetColumns
 - gdcmm::Bitmap, [253](#)
 - gdcmm::Overlay, [893](#)
- GetCommand
 - gdcmm::Subject, [1200](#)
- GetConnectionInfo
 - gdcmm::network::ULConnection, [1375](#)
- GetConstructorString
 - gdcmm::Dicts, [437](#)
- GetContourReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [1546](#)
- GetContourReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [1546](#)
- GetCryptographicMessageSyntax
 - gdcmm::Anonymizer, [140](#)
- GetCSADataInfo
 - gdcmm::CSAHeader, [351](#)
- GetCSAEEnd
 - gdcmm::CSAHeader, [351](#)
- GetCSAElementByName
 - gdcmm::CSAHeader, [352](#)
- GetCSAHeaderDict
 - gdcmm::Dicts, [437](#)
- GetCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, [356](#)
- GetCSAImageHeaderInfoTag
 - gdcmm::CSAHeader, [352](#)
- GetCSASeriesHeaderInfoTag
 - gdcmm::CSAHeader, [352](#)
- GetCTImageSeriesUIDs
 - gdcmm::DirectoryHelper, [448](#)
- GetCurrentByteIndex
 - gdcmm::Parser, [903](#)
- GetCurrentDateTime
 - gdcmm::System, [1233](#)
- GetCurrentModuleFileName
 - gdcmm::System, [1233](#)
- GetCurrentProcessFileName
 - gdcmm::System, [1233](#)
- GetCurrentResourcesDirectory
 - gdcmm::System, [1233](#)
- GetCurve
 - gdcmm::Pixmap, [943](#)
- GetCurveDataDescriptor
 - gdcmm::Curve, [367](#)
- GetCWD
 - gdcmm::System, [1233](#)
- GetData
 - gdcmm::DataEvent, [387](#)
- GetDataElement
 - gdcmm::Bitmap, [253](#)
 - gdcmm::DataSet, [393](#)
 - gdcmm::Item, [724](#)
- GetDataExtraRoot
 - gdcmm::Testing, [1259](#)
- GetDataLength
 - gdcmm::DataEvent, [387](#)

- GetDataRoot
 - gdcmm::Testing, [1260](#)
- GetDataSet
 - gdcmm::CSAHeader, [352](#)
 - gdcmm::DataSetEvent, [403](#)
 - gdcmm::File, [550](#)
- GetDataSetPos
 - gdcmm::network::ULEvent, [1391](#)
- GetDataSets
 - gdcmm::network::ULBasicCallback, [1372](#)
- GetDataSetTransferSyntax
 - gdcmm::FileMetaInformation, [576](#)
- GetDataValueRepresentation
 - gdcmm::Curve, [367](#)
- GetDebugFlag
 - gdcmm::Trace, [1266](#)
- GetDebugStream
 - gdcmm::Trace, [1266](#)
- GetDecodeLength
 - gdcmm::Base64, [223](#)
- GetDEEnd
 - gdcmm::DataSet, [394](#)
- GetDefaultTransferSyntax
 - gdcmm::PresentationContextGenerator, [975](#)
- GetDefs
 - gdcmm::Global, [616](#)
 - gdcmm::TableReader, [1242](#)
- GetDES
 - gdcmm::DataSet, [394](#)
- GetDescription
 - gdcmm::CSAHeaderDictEntry, [358](#)
 - gdcmm::Exception, [537](#)
 - gdcmm::ModuleEntry, [832](#)
 - gdcmm::Overlay, [893](#)
- GetDescriptiveName
 - vtkGDCMImageReader, [1449](#)
 - vtkGDCMImageReader2, [1463](#)
 - vtkGDCMImageWriter, [1476](#)
- GetDict
 - gdcmm::XMLDictReader, [1566](#)
- GetDictEntry
 - gdcmm::Dict, [422](#)
 - gdcmm::Dicts, [437](#)
 - gdcmm::PrivateDict, [988](#)
- GetDictEntryByKeyword
 - gdcmm::Dict, [422](#)
- GetDictEntryByName
 - gdcmm::Dict, [422](#)
- GetDictName
 - gdcmm::DictConverter, [426](#)
- GetDicts
 - gdcmm::Global, [616](#), [617](#)
- GetDictVM
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [162](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [172](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [178](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [198](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [216](#)
- GetDictVR
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [162](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [172](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [178](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [198](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [216](#)
- GetDimension
 - gdcmm::Bitmap, [253](#)
- GetDimensions
 - gdcmm::Bitmap, [253](#)
 - gdcmm::Curve, [367](#)
 - gdcmm::ImageCodec, [662](#)
- GetDimensionsValue
 - gdcmm::ImageHelper, [675](#)
- GetDimensionsValueForResolution
 - gdcmm::StreamImageReader, [1160](#)
- GetDirectionCosines
 - gdcmm::Image, [634](#)
- GetDirectionCosinesFromDataSet
 - gdcmm::ImageHelper, [675](#)
- GetDirectionCosinesTolerance
 - gdcmm::IPPSorter, [717](#)
- GetDirectionCosinesValue

- gdcmm::ImageHelper, 676
- GetDirectories
 - gdcmm::Directory, 446
- GetElapsedTime
 - gdcmm::network::ARTIMTimer, 154
- GetElement
 - gdcmm::Tag, 1248
- GetElementTag
 - gdcmm::Tag, 1248
- GetEncodeLength
 - gdcmm::Base64, 224
- GetErrorCode
 - gdcmm::Parser, 903
- GetErrorFlag
 - gdcmm::Trace, 1266
- GetErrorStream
 - gdcmm::Trace, 1266
- GetErrorString
 - gdcmm::Parser, 903
- GetEvent
 - gdcmm::network::ULEvent, 1391
- GetEventName
 - gdcmm::AnonymizeEvent, 134
 - gdcmm::DataEvent, 387
 - gdcmm::DataSetEvent, 403
 - gdcmm::Event, 534
 - gdcmm::FileNameEvent, 587
 - gdcmm::ProgressEvent, 998
- GetExtension
 - gdcmm::Filename, 582
- GetFactoryInstance
 - gdcmm::CryptoFactory, 337
- GetFile
 - gdcmm::Anonymizer, 140
 - gdcmm::Cleaner, 301
 - gdcmm::DICOMDIRGenerator, 418
 - gdcmm::FileDecompressLookupTable, 563
 - gdcmm::FileDerivation, 566, 567
 - gdcmm::FileExplicitFilter, 570
 - gdcmm::IconImageFilter, 623
 - gdcmm::PythonFilter, 1004
 - gdcmm::Reader, 1027
 - gdcmm::SplitMosaicFilter, 1154
 - gdcmm::StreamImageReader, 1161
 - gdcmm::StringFilter, 1195
 - gdcmm::Writer, 1561
 - vtkGDCMMedicalImageProperties, 1483
- GetFileExtensions
 - vtkGDCMImageReader, 1449
 - vtkGDCMImageReader2, 1463
 - vtkGDCMImageWriter, 1476
- GetFileMetaInformationVersion
 - gdcmm::FileMetaInformation, 577
- GetFileName
 - gdcmm::Filename, 582
 - gdcmm::FileNameEvent, 587
 - gdcmm::Testing, 1260
 - vtkGDCMImageWriter, 1477
 - vtkGDCMThreadedImageReader2, 1505
- GetFilename
 - gdcmm::FilenameGenerator, 589
 - gdcmm::TableReader, 1242
- GetFilenameFromPrivateTagToValue
 - gdcmm::Scanner2, 1064
 - gdcmm::StrictScanner2, 1184
- GetFilenameFromPublicTagToValue
 - gdcmm::Scanner2, 1065
 - gdcmm::StrictScanner2, 1184
- GetFilenameFromTagToValue
 - gdcmm::Scanner, 1054
 - gdcmm::StrictScanner, 1174
- GetFileNames
 - gdcmm::Testing, 1260
- GetFilenames
 - gdcmm::Directory, 446
 - gdcmm::FilenameGenerator, 590
 - gdcmm::Scanner, 1054
 - gdcmm::Scanner2, 1065
 - gdcmm::Sorter, 1146
 - gdcmm::StrictScanner, 1175
 - gdcmm::StrictScanner2, 1184
- GetFilenamesFromSeriesUIDs
 - gdcmm::DirectoryHelper, 448
- GetFiles
 - gdcmm::FileSet, 592
- GetFiniteVolume
 - gdcmm::Surface, 1206
- GetFirstSingleSerieUIDFileSet
 - gdcmm::SerieHelper, 1112
- GetForcePixelSpacing
 - gdcmm::ImageHelper, 676
- GetForceRescaleInterceptSlope
 - gdcmm::ImageHelper, 676
- GetFormat
 - gdcmm::CSAHeader, 352
- GetFragBuffer
 - gdcmm::SequenceOfFragments, 1096
- GetFragment
 - gdcmm::SequenceOfFragments, 1096
- GetFragmentSizeMax
 - gdcmm::ImageFragmentSplitter, 673
- GetFrameOfReference
 - gdcmm::DirectoryHelper, 449
- GetFullLength
 - gdcmm::FileMetaInformation, 577
- GetGDCMDataRoot
 - vtkGDCMTesting, 1496
- GetGDCMImplementationClassUID

- gdcmm::FileMetaInformation, 577
- GetGDCMImplementationVersionName
 - gdcmm::FileMetaInformation, 577
- GetGDCMSourceApplicationEntityTitle
 - gdcmm::FileMetaInformation, 577
- GetGDCMUID
 - gdcmm::UIDGenerator, 1283
- GetGroup
 - gdcmm::Curve, 367
 - gdcmm::Overlay, 893
 - gdcmm::Tag, 1249
- GetHasExpired
 - gdcmm::network::ARTIMTimer, 154
- GetHeader
 - gdcmm::File, 551
- GetHeaderInfo
 - gdcmm::ImageCodec, 662
 - gdcmm::JPEG12Codec, 732
 - gdcmm::JPEG16Codec, 737
 - gdcmm::JPEG2000Codec, 743
 - gdcmm::JPEG8Codec, 750
 - gdcmm::JPEGCodec, 757
 - gdcmm::JPEGLSCodec, 766
 - gdcmm::PGXCodec, 925
 - gdcmm::PNMCodec, 962
 - gdcmm::RAWCodec, 1023
 - gdcmm::RLECodec, 1045
- GetHierarchicalSearchTags
 - gdcmm::QueryBase, 1006
 - gdcmm::QueryImage, 1010
 - gdcmm::QueryPatient, 1013
 - gdcmm::QuerySeries, 1015
 - gdcmm::QueryStudy, 1018
- GetHighBit
 - gdcmm::PixelFormat, 934
- GetHostName
 - gdcmm::System, 1234
- GetIconImage
 - gdcmm::IconImageFilter, 623
 - gdcmm::IconImageGenerator, 626
 - gdcmm::Pixmap, 943
 - vtkGDCMImageReader, 1449
 - vtkGDCMImageReader2, 1463
- GetIconImagePort
 - vtkGDCMImageReader2, 1463
- GetIE
 - gdcmm::IODEntry, 710
- GetImage
 - gdcmm::ImageReader, 683
 - gdcmm::ImageWriter, 696, 697
 - gdcmm::PixmapWriter, 956
 - gdcmm::SplitMosaicFilter, 1154
- GetImplementationClassUID
 - gdcmm::FileMetaInformation, 577
- GetImplementationVersionName
 - gdcmm::FileMetaInformation, 577
- GetIndex
 - gdcmm::SwapCode, 1227
 - gdcmm::VM, 1428
- GetInitialized
 - gdcmm::CAPICryptographicMessageSyntax, 288
- GetInput
 - gdcmm::ImageToImageFilter, 692
 - gdcmm::PixmapToPixmapFilter, 952
 - vtkImageColorViewer, 1514
- GetInputFilename
 - gdcmm::DictConverter, 426
- GetInstance
 - gdcmm::Global, 617
- GetIntercept
 - gdcmm::Image, 634
 - gdcmm::Rescaler, 1038
- GetInterfile
 - gdcmm::CSAHeader, 353
- GetInternal
 - gdcmm::Preamble, 965
- GetIOD
 - gdcmm::IODs, 713
 - gdcmm::SOPClassUIDToIOD, 1142
- GetIODEntry
 - gdcmm::IOD, 708
- GetIODFromFile
 - gdcmm::Defs, 409
- GetIODFromSOPClassUID
 - gdcmm::SOPClassUIDToIOD, 1142
- GetIODNameFromMediaStorage
 - gdcmm::Defs, 409
- GetIODs
 - gdcmm::Defs, 409
- GetIsCommand
 - gdcmm::network::PresentationDataValue, 980
- GetIsLastFragment
 - gdcmm::network::PresentationDataValue, 980
- GetIStream
 - gdcmm::network::ULEvent, 1391
- GetItem
 - gdcmm::SequenceOfItems, 1104, 1105
- GetKey
 - gdcmm::CSAElement, 343
- GetKeys
 - gdcmm::Scanner, 1055
 - gdcmm::Scanner2, 1065
 - gdcmm::StrictScanner, 1175
 - gdcmm::StrictScanner2, 1185
- GetKeyword
 - gdcmm::DictEntry, 429
- GetKeywordFromTag
 - gdcmm::Dict, 422

- GetLabel
 - gdcm::Orientation, [888](#)
- GetLastElement
 - gdcm::ParseException, [900](#)
- GetLastSystemError
 - gdcm::System, [1234](#)
- GetLength
 - gdcm::ByteValue, [279](#)
 - gdcm::CP246ExplicitDataElement, [334](#)
 - gdcm::DataElement, [374](#)
 - gdcm::DataSet, [394](#)
 - gdcm::Element< TVR, TVM >, [458](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [465](#)
 - gdcm::Element< TVR, VM::VM1_n >, [470](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [478](#)
 - gdcm::Element< TVR, VM::VM2_n >, [483](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [490](#)
 - gdcm::Element< TVR, VM::VM3_4 >, [495](#)
 - gdcm::Element< TVR, VM::VM3_n >, [501](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [505](#)
 - gdcm::Element< VR::OB, VM::VM1 >, [510](#)
 - gdcm::Element< VR::OW, VM::VM1 >, [515](#)
 - gdcm::ExplicitDataElement, [542](#)
 - gdcm::ExplicitImplicitDataElement, [546](#)
 - gdcm::Fragment, [613](#)
 - gdcm::ImplicitDataElement, [704](#)
 - gdcm::Item, [724](#)
 - gdcm::Preamble, [965](#)
 - gdcm::SequenceOfFragments, [1096](#)
 - gdcm::SequenceOfItems, [1105](#)
 - gdcm::Tag, [1249](#)
 - gdcm::UNExplicitDataElement, [1398](#)
 - gdcm::UNExplicitImplicitDataElement, [1402](#)
 - gdcm::Value, [1416](#)
 - gdcm::VL, [1422](#)
 - gdcm::VM, [1428](#)
 - gdcm::VR, [1434](#)
 - gdcm::VR16ExplicitDataElement, [1440](#)
- GetLocaleCharset
 - gdcm::System, [1234](#)
- GetLossless
 - gdcm::JPEGCodec, [757](#)
 - gdcm::JPEGLSCodec, [766](#)
- GetLossyFlag
 - gdcm::ImageCodec, [662](#)
- GetLossyFlagFromFile
 - gdcm::Testing, [1260](#)
- GetLUT
 - gdcm::Bitmap, [254](#)
 - gdcm::ImageCodec, [662](#)
 - gdcm::ImageHelper, [676](#)
 - gdcm::LookupTable, [781](#)
- GetLUTDescriptor
 - gdcm::LookupTable, [781](#)
- GetLUTLength
 - gdcm::LookupTable, [781](#)
- GetMacro
 - gdcm::Macros, [791](#)
- GetMacroEntry
 - gdcm::Macro, [788](#)
- GetMacros
 - gdcm::Defs, [409](#)
- GetMajorAxisFromPatientRelativeDirectionCosine
 - gdcm::Orientation, [888](#)
- GetMajorVersion
 - gdcm::Version, [1419](#)
- GetManifold
 - gdcm::Surface, [1206](#)
- GetMapping
 - gdcm::Scanner, [1055](#)
 - gdcm::StrictScanner, [1175](#)
- GetMappingFromPrivateTagToValue
 - gdcm::Scanner2, [1065](#)
 - gdcm::StrictScanner2, [1185](#)
- GetMappingFromPublicTagToValue
 - gdcm::Scanner2, [1065](#)
 - gdcm::StrictScanner2, [1185](#)
- GetMappingFromTagToValue
 - gdcm::Scanner, [1055](#)
 - gdcm::StrictScanner, [1175](#)
- GetMappings
 - gdcm::Scanner, [1055](#)
 - gdcm::StrictScanner, [1175](#)
- GetMax
 - gdcm::PixelFormat, [934](#)
- GetMaximumLength
 - gdcm::network::MaximumLengthSub, [792](#)
- GetMaximumLengthSub
 - gdcm::network::UserInformation, [1409](#)
- GetMaximumPointDistance
 - gdcm::Surface, [1206](#)
- GetMaxLength
 - gdcm::PersonName, [920](#)
- GetMaxPDULength
 - gdcm::network::ULConnectionInfo, [1381](#)
- GetMaxPDUSize
 - gdcm::network::ULConnection, [1375](#)
- GetMD5DataImage
 - gdcm::Testing, [1260](#)
- GetMD5DataImages
 - gdcm::Testing, [1261](#)
- GetMD5FromBrokenFile
 - gdcm::Testing, [1261](#)
- GetMD5FromFile
 - gdcm::Testing, [1261](#)
- GetMD5MetaImage
 - vtkGDCMTesting, [1496](#)
- GetMeanPointDistance

- gdcmm::Surface, [1206](#)
- GetMediaStorage
 - gdcmm::DataSet, [395](#)
 - gdcmm::FileMetaInformation, [577](#)
- GetMediaStorageAsString
 - gdcmm::FileMetaInformation, [578](#)
- GetMediaStorageDataFile
 - gdcmm::Testing, [1261](#)
- GetMediaStorageDataFiles
 - gdcmm::Testing, [1261](#)
- GetMediaStorageFromFile
 - gdcmm::Testing, [1261](#)
- GetMeshPrimitive
 - gdcmm::Surface, [1206](#), [1207](#)
- GetMessageHeader
 - gdcmm::network::PresentationDataValue, [980](#)
- GetMetaInformationTS
 - gdcmm::FileMetaInformation, [578](#)
- GetMHDMD5FromFile
 - vtkGDCMTesting, [1496](#)
- GetMin
 - gdcmm::PixelFormat, [934](#)
- GetMinorVersion
 - gdcmm::Version, [1420](#)
- GetModality
 - gdcmm::MediaStorage, [802](#)
- GetModalityDimension
 - gdcmm::MediaStorage, [802](#)
- GetModule
 - gdcmm::Modules, [835](#)
- GetModuleEntry
 - gdcmm::NestedModuleEntries, [857](#)
- GetModuleEntryInMacros
 - gdcmm::Module, [828](#)
- GetModules
 - gdcmm::Defs, [409](#), [410](#)
- GetMPType
 - gdcmm::MeshPrimitive, [815](#)
- GetMPTypeString
 - gdcmm::MeshPrimitive, [815](#)
- GetMRIImageSeriesUIDs
 - gdcmm::DirectoryHelper, [449](#)
- GetMrProtocol
 - gdcmm::CSAHeader, [353](#)
- GetMrProtocolByName
 - gdcmm::MrProtocol, [845](#)
- GetMSString
 - gdcmm::MediaStorage, [802](#)
- GetMSType
 - gdcmm::MediaStorage, [802](#)
- GetMTime
 - vtkImageMapToColors16, [1525](#)
- GetName
 - gdcmm::CSAElement, [343](#)
 - gdcmm::CSAHeaderDictEntry, [358](#)
 - gdcmm::DictEntry, [429](#)
 - gdcmm::Filename, [582](#)
 - gdcmm::GroupDict, [620](#)
 - gdcmm::IODEntry, [710](#)
 - gdcmm::Macro, [788](#)
 - gdcmm::Module, [828](#)
 - gdcmm::ModuleEntry, [832](#)
 - gdcmm::network::AbstractSyntax, [130](#)
 - gdcmm::network::ApplicationContext, [146](#)
 - gdcmm::network::TransferSyntaxSub, [1276](#)
 - gdcmm::PDBelement, [909](#)
 - gdcmm::QueryBase, [1007](#)
 - gdcmm::QueryImage, [1010](#)
 - gdcmm::QueryPatient, [1013](#)
 - gdcmm::QuerySeries, [1015](#)
 - gdcmm::QueryStudy, [1018](#)
 - gdcmm::UIDs, [1327](#)
- GetNeedByteSwap
 - gdcmm::Bitmap, [254](#)
 - gdcmm::ImageCodec, [663](#)
- GetNegotiatedType
 - gdcmm::TransferSyntax, [1273](#)
- GetNestedDataSet
 - gdcmm::Item, [724](#)
- GetNextSingleSerieUIDFileSet
 - gdcmm::SerieHelper, [1112](#)
- GetNoOfItems
 - gdcmm::CSAElement, [344](#)
- GetNumberOfComponents
 - gdcmm::PersonName, [920](#)
- GetNumberOfContourReferencedFrameOfReferences
 - vtkRTStructSetProperties, [1546](#), [1547](#)
- GetNumberOfCurves
 - gdcmm::Curve, [367](#)
 - gdcmm::Pixmap, [944](#)
- GetNumberOfDimensions
 - gdcmm::Bitmap, [254](#)
 - gdcmm::ImageCodec, [663](#)
- GetNumberOfElementsFromArray
 - gdcmm::VM, [1428](#)
- GetNumberOfFileNames
 - gdcmm::Testing, [1261](#)
- GetNumberOfFileNames
 - gdcmm::FilenameGenerator, [590](#)
- GetNumberOfFragments
 - gdcmm::SequenceOfFragments, [1097](#)
- GetNumberOfIconImages
 - gdcmm::IconImageFilter, [623](#)
- GetNumberOfImagesInMosaic
 - gdcmm::SplitMosaicFilter, [1154](#)
- GetNumberOfIODs
 - gdcmm::IOD, [708](#)
- GetNumberOfItems

- gdcmm::SequenceOfItems, [1105](#)
- GetNumberOfMD5DataImages
 - gdcmm::Testing, [1262](#)
- GetNumberOfMD5MetaImages
 - vtkGDCMTesting, [1496](#)
- GetNumberOfMediaStorageDataFiles
 - gdcmm::Testing, [1262](#)
- GetNumberOfModality
 - gdcmm::MediaStorage, [803](#)
- GetNumberOfModuleEntries
 - gdcmm::NestedModuleEntries, [857](#)
- GetNumberOfMSString
 - gdcmm::MediaStorage, [803](#)
- GetNumberOfMSType
 - gdcmm::MediaStorage, [803](#)
- GetNumberOfOverlays
 - gdcmm::Pixmap, [944](#)
- GetNumberOfPoints
 - gdcmm::Curve, [367](#)
- GetNumberOfPresentationContext
 - gdcmm::network::AAssociateRQPDU, [125](#)
- GetNumberOfPresentationContextAC
 - gdcmm::network::AAssociateACPDU, [118](#)
- GetNumberOfPresentationDataValues
 - gdcmm::network::PDataTFPDU, [906](#)
- GetNumberOfPrimitivesData
 - gdcmm::MeshPrimitive, [815](#)
- GetNumberOfReferencedFrameOfReferences
 - vtkRTStructSetProperties, [1547](#)
- GetNumberOfSegments
 - gdcmm::SegmentWriter, [1090](#)
- GetNumberOfSOPClassToIOD
 - gdcmm::SOPClassUIDToIOD, [1143](#)
- GetNumberOfStructureSetROIs
 - vtkRTStructSetProperties, [1547](#)
- GetNumberOfSurfacePoints
 - gdcmm::Surface, [1207](#)
- GetNumberOfSurfaces
 - gdcmm::SurfaceReader, [1220](#)
 - gdcmm::SurfaceWriter, [1224](#)
- GetNumberOfTransferSyntaxes
 - gdcmm::network::PresentationContextRQ, [977](#)
 - gdcmm::PresentationContext, [969](#)
- GetNumberOfTransferSyntaxStrings
 - gdcmm::UIDs, [1327](#)
- GetNumberOfValues
 - gdcmm::Attribute< Group, Element, TVR, TVM
>, [162](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_n >, [172](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_3 >, [178](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_n >, [190](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_2n >, [198](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_n >, [216](#)
- GetNumberOfVectors
 - gdcmm::Surface, [1207](#)
- GetObliquityThresholdCosineValue
 - gdcmm::Orientation, [888](#)
- GetOffScreenRendering
 - vtkImageColorViewer, [1514](#)
- GetOptionalTags
 - gdcmm::QueryBase, [1007](#)
 - gdcmm::QueryImage, [1011](#)
 - gdcmm::QueryPatient, [1013](#)
 - gdcmm::QuerySeries, [1016](#)
 - gdcmm::QueryStudy, [1018](#)
- GetOrderedValues
 - gdcmm::Scanner, [1055](#)
 - gdcmm::StrictScanner, [1175](#)
- GetOrigin
 - gdcmm::Image, [634](#)
 - gdcmm::Overlay, [894](#)
- GetOriginValue
 - gdcmm::ImageHelper, [676](#)
- GetOuput
 - gdcmm::ImageConverter, [669](#)
- GetOutput
 - gdcmm::BitmapToBitmapFilter, [264](#)
 - gdcmm::ImageToImageFilter, [692](#)
 - gdcmm::PixmapToPixmapFilter, [952](#)
- GetOutputAsBitmap
 - gdcmm::BitmapToBitmapFilter, [264](#)
- GetOutputAsPixmap
 - gdcmm::PixmapToPixmapFilter, [952](#)
- GetOutputFilename
 - gdcmm::DictConverter, [426](#)
- GetOutputType
 - gdcmm::DictConverter, [426](#)
- GetOverlay
 - gdcmm::Pixmap, [944](#)
 - vtkGDCMImageReader, [1449](#)
 - vtkGDCMImageReader2, [1463](#)
- GetOverlayData
 - gdcmm::Overlay, [894](#)
- GetOverlayPort
 - vtkGDCMImageReader2, [1463](#)
- GetOverlayTypeAsString
 - gdcmm::Overlay, [894](#)

- GetOverlayTypeFromString
 - gdcm::Overlay, [894](#)
- GetOverlayVisibility
 - vtkImageColorViewer, [1514](#)
- GetOwner
 - gdcm::PrivateTag, [993](#)
- GetPath
 - gdcm::Filename, [583](#)
- GetPattern
 - gdcm::FilenameGenerator, [590](#)
- GetPDBEEnd
 - gdcm::PDBHeader, [912](#)
- GetPDBElementByName
 - gdcm::PDBHeader, [912](#)
- GetPDBInfoTag
 - gdcm::PDBHeader, [913](#)
- GetPDUs
 - gdcm::network::ULEvent, [1391](#)
- GetPDVs
 - gdcm::network::PDUFactory, [919](#)
- GetPermissions
 - gdcm::System, [1234](#)
- GetPhotometricInterpretation
 - gdcm::Bitmap, [254](#)
 - gdcm::ImageChangePhotometricInterpretation, [643](#)
 - gdcm::ImageCodec, [663](#)
- GetPhotometricInterpretationValue
 - gdcm::ImageHelper, [676](#)
- GetPIString
 - gdcm::PhotometricInterpretation, [928](#)
- GetPIType
 - gdcm::PhotometricInterpretation, [928](#)
- GetPixelFormat
 - gdcm::Bitmap, [255](#)
 - gdcm::ImageCodec, [663](#)
- GetPixelFormatValue
 - gdcm::ImageHelper, [676](#)
- GetPixelRepresentation
 - gdcm::PixelFormat, [934](#)
- GetPixelSize
 - gdcm::PixelFormat, [934](#)
- GetPixelSpacingDataRoot
 - gdcm::Testing, [1262](#)
- GetPixmap
 - gdcm::FileDecompressLookupTable, [564](#)
 - gdcm::IconImageGenerator, [626](#)
 - gdcm::PixmapReader, [949](#)
 - gdcm::PixmapWriter, [956](#)
- GetPlanarConfiguration
 - gdcm::Bitmap, [255](#)
 - gdcm::ImageChangePlanarConfiguration, [648](#)
 - gdcm::ImageCodec, [663](#)
- GetPlanarConfigurationValue
 - gdcm::ImageHelper, [677](#)
- GetPMSRescaleInterceptSlope
 - gdcm::ImageHelper, [677](#)
- GetPMTFInformationDataTag
 - gdcm::MEC_MR3, [794](#)
- GetPointCoordinatesData
 - gdcm::Surface, [1207](#)
- GetPointer
 - gdcm::ByteValue, [279](#)
 - gdcm::LookupTable, [781](#)
 - gdcm::SmartPointer< ObjectType >, [1139](#)
 - vtkLookupTable16, [1541](#)
- GetPointerFromElement
 - gdcm::ImageHelper, [677](#)
- GetPointPositionAccuracy
 - gdcm::Surface, [1207](#)
- GetPointsBoundingBoxCoordinates
 - gdcm::Surface, [1207](#)
- GetPosition
 - vtkImageColorViewer, [1514](#)
- GetPreamble
 - gdcm::FileMetaInformation, [578](#)
- GetPrefix
 - gdcm::FilenameGenerator, [590](#)
- GetPresentationContext
 - gdcm::network::AAssociateRQPDU, [125](#)
- GetPresentationContextAC
 - gdcm::network::AAssociateACPDU, [118](#)
- GetPresentationContextACByID
 - gdcm::network::ULConnection, [1375](#)
- GetPresentationContextByAbstractSyntax
 - gdcm::network::AAssociateRQPDU, [125](#)
- GetPresentationContextByID
 - gdcm::network::AAssociateRQPDU, [125](#)
- GetPresentationContextID
 - gdcm::network::PresentationContextAC, [971](#)
 - gdcm::network::PresentationContextRQ, [977](#)
 - gdcm::network::PresentationDataValue, [980](#)
 - gdcm::PresentationContext, [969](#)
- GetPresentationContextIDFromPresentationContext
 - gdcm::network::ULConnection, [1375](#)
- GetPresentationContextRQByID
 - gdcm::network::ULConnection, [1375](#)
- GetPresentationContexts
 - gdcm::network::AAssociateRQPDU, [125](#)
 - gdcm::network::ULConnection, [1375](#)
 - gdcm::PresentationContextGenerator, [975](#)
- GetPresentationDataValue
 - gdcm::network::PDataTFPDU, [906](#)
- GetPrettyPrint
 - gdcm::JSON, [768](#)
- GetPrimitiveData
 - gdcm::MeshPrimitive, [816](#)
- GetPrimitivesData

- gdcmm::MeshPrimitive, [816](#)
- GetPrimitiveType
 - gdcmm::MeshPrimitive, [816](#)
- GetPrintStyle
 - gdcmm::Printer, [985](#)
 - gdcmm::XMLPrinter, [1569](#)
- GetPrivateCreator
 - gdcmm::DataSet, [395](#)
 - gdcmm::Tag, [1249](#)
- GetPrivateDict
 - gdcmm::Dicts, [437](#)
 - gdcmm::XMLPrivateDictReader, [1573](#)
- GetPrivateMapping
 - gdcmm::Scanner2, [1065](#)
 - gdcmm::StrictScanner2, [1185](#)
- GetPrivateMappings
 - gdcmm::Scanner2, [1065](#)
 - gdcmm::StrictScanner2, [1185](#)
- GetPrivateOrderedValues
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1185](#)
- GetPrivateTag
 - gdcmm::DataSet, [395](#)
- GetPrivateValue
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1185](#)
- GetPrivateValues
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1186](#)
- GetProcessingAlgorithm
 - gdcmm::Surface, [1207](#), [1208](#)
- GetProgress
 - gdcmm::ProgressEvent, [998](#)
- GetPropertyCategory
 - gdcmm::Segment, [1073](#)
- GetPropertyType
 - gdcmm::Segment, [1073](#)
- GetPropertyTypeModifiers
 - gdcmm::Segment, [1074](#)
- GetProtocol
 - gdcmm::network::ULConnection, [1375](#)
- GetPublicDict
 - gdcmm::Dicts, [438](#)
- GetPublicMapping
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1186](#)
- GetPublicMappings
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1186](#)
- GetPublicOrderedValues
 - gdcmm::Scanner2, [1066](#)
 - gdcmm::StrictScanner2, [1186](#)
- GetPublicValue
 - gdcmm::Scanner2, [1066](#)
- gdcmm::StrictScanner2, [1186](#)
- GetPublicValues
 - gdcmm::Scanner2, [1067](#)
 - gdcmm::StrictScanner2, [1186](#)
- GetQuality
 - gdcmm::JPEG2000Codec, [743](#)
 - gdcmm::JPEGCodec, [757](#)
- GetQueryDataSet
 - gdcmm::BaseQuery, [234](#)
- GetQueryLevel
 - gdcmm::QueryBase, [1007](#)
 - gdcmm::QueryImage, [1011](#)
 - gdcmm::QueryPatient, [1013](#)
 - gdcmm::QuerySeries, [1016](#)
 - gdcmm::QueryStudy, [1018](#)
- GetQueryLevelFromQueryRoot
 - gdcmm::BaseRootQuery, [239](#)
- GetQueryLevelFromString
 - gdcmm::BaseRootQuery, [239](#)
- GetQueryLevelString
 - gdcmm::BaseRootQuery, [239](#)
- GetRate
 - gdcmm::JPEG2000Codec, [743](#)
- GetRAWMD5FromFile
 - vtkGDCMTesting, [1497](#)
- GetRealWorldValueMappingContent
 - gdcmm::ImageHelper, [677](#)
- GetReason
 - gdcmm::network::PresentationContextAC, [971](#)
- GetRecommendedDisplayCIELabValue
 - gdcmm::Surface, [1208](#)
- GetRecommendedDisplayGrayscaleValue
 - gdcmm::Surface, [1208](#)
- GetRecommendedPresentationOpacity
 - gdcmm::Surface, [1208](#)
- GetRecommendedPresentationType
 - gdcmm::Surface, [1208](#)
- GetRef
 - gdcmm::IODEntry, [710](#)
- GetReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [1547](#)
- GetReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [1547](#)
- GetRegion
 - gdcmm::ImageRegionReader, [689](#)
- GetRequiredDataSet
 - gdcmm::ModalityPerformedProcedureStepCreate-Query, [820](#)
 - gdcmm::ModalityPerformedProcedureStepSet-Query, [824](#)
- GetRequiredTags
 - gdcmm::QueryBase, [1007](#)
 - gdcmm::QueryImage, [1011](#)
 - gdcmm::QueryPatient, [1013](#)

- gdcmm::QuerySeries, [1016](#)
- gdcmm::QueryStudy, [1018](#)
- GetRescaleInterceptSlopeValue
 - gdcmm::ImageHelper, [677](#)
- GetReserved43_74
 - gdcmm::network::AAssociateRQPDU, [125](#)
- GetResponses
 - gdcmm::network::ULBasicCallback, [1372](#)
- GetRetired
 - gdcmm::DictEntry, [429](#)
- GetRoot
 - gdcmm::UIDGenerator, [1284](#)
- GetRows
 - gdcmm::Bitmap, [255](#)
 - gdcmm::Overlay, [894](#)
- GetRTStructSeriesUIDs
 - gdcmm::DirectoryHelper, [449](#)
- GetSamplesPerPixel
 - gdcmm::PhotometricInterpretation, [928](#)
 - gdcmm::PixelFormat, [935](#)
- GetScalarType
 - gdcmm::PixelFormat, [935](#)
- GetScalarTypeAsString
 - gdcmm::PixelFormat, [935](#)
- GetScanner
 - gdcmm::DICOMDIRGenerator, [418](#)
- GetSecondaryCaptureImagePlaneModule
 - gdcmm::ImageHelper, [677](#)
- GetSegment
 - gdcmm::SegmentWriter, [1090](#)
- GetSegmentAlgorithmName
 - gdcmm::Segment, [1074](#)
- GetSegmentAlgorithmType
 - gdcmm::Segment, [1074](#)
- GetSegmentDescription
 - gdcmm::Segment, [1074](#)
- GetSegmentLabel
 - gdcmm::Segment, [1074](#)
- GetSegmentNumber
 - gdcmm::Segment, [1074](#)
- GetSegments
 - gdcmm::SegmentReader, [1085](#)
 - gdcmm::SegmentWriter, [1090](#)
- GetSelectedPrivateGroupOffsetFromFile
 - gdcmm::Testing, [1262](#)
- GetSelectedTagsOffsetFromFile
 - gdcmm::Testing, [1262](#)
- GetSequenceOfFragments
 - gdcmm::DataElement, [374](#)
- GetSeriesUIDsBySOPClassUID
 - gdcmm::DirectoryHelper, [449](#)
- GetSize
 - gdcmm::VR, [1434](#)
 - vtkImageColorViewer, [1514](#)
- GetSizeof
 - gdcmm::VR, [1434](#)
- GetSliceArray
 - gdcmm::MrProtocol, [845](#)
- GetSliceMax
 - vtkImageColorViewer, [1515](#)
- GetSliceMin
 - vtkImageColorViewer, [1515](#)
- GetSliceRange
 - vtkImageColorViewer, [1515](#)
- GetSlope
 - gdcmm::Image, [634](#)
 - gdcmm::Rescaler, [1038](#)
- GetSOPClassUID
 - gdcmm::DirectoryHelper, [449](#)
- GetSOPClassUIDFromIOD
 - gdcmm::SOPClassUIDToIOD, [1143](#)
- GetSOPClassUIDToIOD
 - gdcmm::SOPClassUIDToIOD, [1143](#)
- GetSOPClassUIDToIODs
 - gdcmm::SOPClassUIDToIOD, [1143](#)
- GetSOPInstanceUID
 - gdcmm::BaseQuery, [234](#)
- GetSourceApplicationEntityTitle
 - gdcmm::FileMetaInformation, [578](#)
- GetSourceDirectory
 - gdcmm::Testing, [1262](#)
- GetSpacing
 - gdcmm::Image, [634](#)
- GetSpacingTagFromMediaStorage
 - gdcmm::ImageHelper, [678](#)
- GetSpacingValue
 - gdcmm::ImageHelper, [678](#)
- GetStart
 - gdcmm::ByteBuffer, [271](#)
- GetState
 - gdcmm::network::ULConnection, [1376](#)
- GetStateIndex
 - gdcmm::network, [108](#)
- GetSTATES
 - gdcmm::Surface, [1208](#)
- GetSTATESString
 - gdcmm::Surface, [1208](#)
- GetStream
 - gdcmm::Trace, [1266](#)
- GetStreamCurrentPosition
 - gdcmm::Reader, [1028](#)
- GetStreamOffsetFromFile
 - gdcmm::Testing, [1262](#)
- GetStreamPtr
 - gdcmm::Reader, [1028](#)
 - gdcmm::Writer, [1562](#)
- GetString
 - gdcmm::MediaStorage, [803](#)

- gdcmm::PhotometricInterpretation, 929
- gdcmm::TransferSyntax, 1273
- gdcmm::UIDs, 1328
- GetStringValueFromTag
 - gdcmm::DirectoryHelper, 449
- GetStructureSetObservationNumber
 - vtkRTStructSetProperties, 1547
- GetStructureSetROIDescription
 - vtkRTStructSetProperties, 1547
- GetStructureSetROIGenerationAlgorithm
 - vtkRTStructSetProperties, 1547
- GetStructureSetROIName
 - vtkRTStructSetProperties, 1547
- GetStructureSetROINumber
 - vtkRTStructSetProperties, 1548
- GetStructureSetROIObservationLabel
 - vtkRTStructSetProperties, 1548
- GetStructureSetROIRefFrameRefUID
 - vtkRTStructSetProperties, 1548
- GetStructureSetRTROIInterpretedType
 - vtkRTStructSetProperties, 1548
- GetSurface
 - gdcmm::Segment, 1074
- GetSurfaceComments
 - gdcmm::Surface, 1208
- GetSurfaceCount
 - gdcmm::Segment, 1074
- GetSurfaceNumber
 - gdcmm::Surface, 1209
- GetSurfaceProcessing
 - gdcmm::Surface, 1209
- GetSurfaceProcessingDescription
 - gdcmm::Surface, 1209
- GetSurfaceProcessingRatio
 - gdcmm::Surface, 1209
- GetSurfaces
 - gdcmm::Segment, 1075
- GetSwapCode
 - gdcmm::TransferSyntax, 1273
- GetSwapCodeString
 - gdcmm::SwapCode, 1227
- GetSyngoDT
 - gdcmm::CSAElement, 344
- GetTable
 - gdcmm::SequenceOfFragments, 1097
- GetTableEntry
 - gdcmm::Table, 1238
- GetTag
 - gdcmm::AnonymizeEvent, 134
 - gdcmm::Attribute< Group, Element, TVR, TVM
>, 162
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1 >, 172
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_3 >, 179
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_8 >, 184
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM1_n >, 190
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_2n >, 198
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM2_n >, 204
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_3n >, 211
 - gdcmm::Attribute< Group, Element, TVR,
VM::VM3_n >, 217
 - gdcmm::DataElement, 374, 375
- GetTagListByLevel
 - gdcmm::BaseRootQuery, 239
 - gdcmm::FindPatientRootQuery, 604
 - gdcmm::FindStudyRootQuery, 608
 - gdcmm::MovePatientRootQuery, 838
 - gdcmm::MoveStudyRootQuery, 842
 - gdcmm::WLMFindQuery, 1557
- GetTempDirectory
 - gdcmm::Testing, 1263
- GetTempDirectoryW
 - gdcmm::Testing, 1263
- GetTempFilename
 - gdcmm::Testing, 1263
- GetTempFilenameW
 - gdcmm::Testing, 1263
- GetTimeout
 - gdcmm::network::ARTIMTimer, 155
 - gdcmm::ServiceClassUser, 1119
- GetTimer
 - gdcmm::network::ULConnection, 1376
- GetTimezoneOffsetFromUTC
 - gdcmm::System, 1234
- GetToplevel
 - gdcmm::Directory, 446
- GetToshibaMECMR3Tag
 - gdcmm::MEC_MR3, 794
- GetTransferSyntax
 - gdcmm::Bitmap, 255
 - gdcmm::ImageChangeTransferSyntax, 653
 - gdcmm::network::PresentationContextAC, 971
 - gdcmm::network::PresentationContextRQ, 977
 - gdcmm::PresentationContext, 969
- GetTransferSyntaxes
 - gdcmm::network::PresentationContextRQ, 978
- GetTransferSyntaxString
 - gdcmm::UIDs, 1328
- GetTransferSyntaxStrings
 - gdcmm::UIDs, 1328
- GetTSSString

- gdcmm::TransferSyntax, [1273](#)
- GetTSType
 - gdcmm::TransferSyntax, [1273](#)
- GetType
 - gdcmm::ModuleEntry, [832](#)
 - gdcmm::Orientation, [888](#)
 - gdcmm::Overlay, [894](#)
 - gdcmm::PhotometricInterpretation, [929](#)
- GetTypeAsEnum
 - gdcmm::Overlay, [894](#)
- GetTypeFromTag
 - gdcmm::Defs, [410](#)
 - gdcmm::IOD, [708](#)
- GetTypeOfData
 - gdcmm::Curve, [367](#)
- GetTypeOfDataDescription
 - gdcmm::Curve, [367](#)
- GetTypeString
 - gdcmm::Type, [1281](#)
- GetTypeType
 - gdcmm::Type, [1281](#)
- GetUIDName
 - gdcmm::UIDs, [1328](#)
- GetUIDString
 - gdcmm::UIDs, [1328](#)
- GetUniqueTags
 - gdcmm::QueryBase, [1007](#)
 - gdcmm::QueryImage, [1011](#)
 - gdcmm::QueryPatient, [1014](#)
 - gdcmm::QuerySeries, [1016](#)
 - gdcmm::QueryStudy, [1019](#)
- GetUnpackBuffer
 - gdcmm::Overlay, [894](#)
- GetUnpackBufferLength
 - gdcmm::Overlay, [895](#)
- GetUsage
 - gdcmm::IODEntry, [710](#)
- GetUsageString
 - gdcmm::Usage, [1406](#)
- GetUsageType
 - gdcmm::IODEntry, [710](#)
 - gdcmm::Usage, [1406](#)
- GetUserData
 - gdcmm::Parser, [903](#)
- GetUserInformation
 - gdcmm::network::AAssociateACPDU, [118](#)
 - gdcmm::network::AAssociateRQPDU, [125](#)
- GetValidatedFile
 - gdcmm::Validate, [1413](#)
- GetValidDataSet
 - gdcmm::WLMFindQuery, [1557](#)
- GetValue
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [163](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [172](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [190](#), [191](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
- gdcmm::CSAElement, [344](#)
- gdcmm::DataElement, [375](#)
- gdcmm::Element< TVR, TVM >, [459](#)
- gdcmm::Element< TVR, VM::VM1_2 >, [465](#)
- gdcmm::Element< TVR, VM::VM1_n >, [470](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
- gdcmm::Element< TVR, VM::VM2_n >, [483](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
- gdcmm::Element< TVR, VM::VM3_4 >, [495](#)
- gdcmm::Element< TVR, VM::VM3_n >, [501](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [505](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [515](#)
- gdcmm::PDBElement, [909](#)
- gdcmm::Scanner, [1055](#)
- gdcmm::StrictScanner, [1176](#)
- GetValueAsSQ
 - gdcmm::DataElement, [375](#)
- GetValues
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [163](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [172](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [191](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
 - gdcmm::Element< TVR, TVM >, [459](#)

- gdcmm::Element< TVR, VM::VM1_2 >, [465](#)
- gdcmm::Element< TVR, VM::VM1_n >, [470](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
- gdcmm::Element< TVR, VM::VM2_n >, [483](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
- gdcmm::Element< TVR, VM::VM3_4 >, [495](#)
- gdcmm::Element< TVR, VM::VM3_n >, [501](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [505](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [515](#)
- gdcmm::Scanner, [1056](#)
- gdcmm::Scanner2, [1067](#)
- gdcmm::StrictScanner, [1176](#)
- gdcmm::StrictScanner2, [1187](#)
- GetVectorAccuracy
 - gdcmm::Surface, [1209](#)
- GetVectorCoordinateData
 - gdcmm::Surface, [1209](#)
- GetVectorDimensionality
 - gdcmm::Surface, [1209](#)
- GetVersion
 - gdcmm::MrProtocol, [845](#)
 - gdcmm::Version, [1420](#)
- GetVIEWType
 - gdcmm::Surface, [1209](#)
- GetVIEWTypeString
 - gdcmm::Surface, [1209](#)
- GetVL
 - gdcmm::DataElement, [376](#)
- GetVL16Max
 - gdcmm::VL, [1422](#)
- GetVL32Max
 - gdcmm::VL, [1422](#)
- GetVM
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [163](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [191](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
 - gdcmm::CSAElement, [344](#)
 - gdcmm::CSAHeaderDictEntry, [359](#)
- gdcmm::DictEntry, [429](#)
- gdcmm::Element< TVR, TVM >, [459](#)
- gdcmm::Element< TVR, VM::VM1_2 >, [465](#)
- gdcmm::Element< TVR, VM::VM1_n >, [470](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
- gdcmm::Element< TVR, VM::VM2_n >, [483](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
- gdcmm::Element< TVR, VM::VM3_4 >, [495](#)
- gdcmm::Element< TVR, VM::VM3_n >, [501](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [505](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [515](#)
- GetVMString
 - gdcmm::VM, [1428](#)
- GetVMType
 - gdcmm::VM, [1429](#)
- GetVMTypeFromLength
 - gdcmm::VM, [1429](#)
- GetVoidPointer
 - gdcmm::ByteValue, [280](#)
- GetVR
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [164](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [184](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [191](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [204](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
 - gdcmm::CSAElement, [345](#)
 - gdcmm::CSAHeaderDictEntry, [359](#)
 - gdcmm::DataElement, [376](#)
 - gdcmm::DictEntry, [429](#)
 - gdcmm::Element< TVR, TVM >, [459](#)
 - gdcmm::Element< TVR, VM::VM1_2 >, [466](#)
 - gdcmm::Element< TVR, VM::VM1_n >, [470](#)
 - gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
 - gdcmm::Element< TVR, VM::VM2_n >, [484](#)
 - gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
 - gdcmm::Element< TVR, VM::VM3_4 >, [496](#)
 - gdcmm::Element< TVR, VM::VM3_n >, [502](#)
 - gdcmm::Element< VR::AS, VM::VM5 >, [505](#)
 - gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
 - gdcmm::Element< VR::OW, VM::VM1 >, [515](#)

- GetVRFromTag
 - gdcm, [92](#)
- GetVRString
 - gdcm::VR, [1435](#)
- GetVRStringFromFile
 - gdcm::VR, [1435](#)
- GetVRType
 - gdcm::VR, [1435](#)
- GetVRTypeFromFile
 - gdcm::VR, [1435](#)
- GetVTKDataRoot
 - vtkGDCMTesting, [1497](#)
- GetWarningFlag
 - gdcm::Trace, [1266](#)
- GetWarningStream
 - gdcm::Trace, [1267](#)
- GetWindowName
 - vtkImageColorViewer, [1515](#)
- GetXMax
 - gdcm::BoxRegion, [268](#)
- GetXMin
 - gdcm::BoxRegion, [268](#)
- GetYMax
 - gdcm::BoxRegion, [268](#)
- GetYMin
 - gdcm::BoxRegion, [269](#)
- GetZMax
 - gdcm::BoxRegion, [269](#)
- GetZMin
 - gdcm::BoxRegion, [269](#)
- GetZSpacing
 - gdcm::IPPSorter, [717](#)
- GetZSpacingTagFromMediaStorage
 - gdcm::ImageHelper, [678](#)
- GetZSpacingTolerance
 - gdcm::IPPSorter, [717](#)
- Global
 - gdcm::Defs, [411](#)
 - gdcm::Dicts, [438](#)
 - gdcm::Global, [616](#)
- GlobalInstance
 - gdcm, [102](#)
- GrabOverlayFromPixelData
 - gdcm::Overlay, [895](#)
- Graphics
 - gdcm::Overlay, [892](#)
- GRAY
 - gdcm::LookupTable, [779](#)
- GrayscalePlanarMPRVolumetricPresentationStateStorage
 - age
 - gdcm::UIDs, [1309](#)
- GrayscaleSoftcopyPresentationStateStorageSOP-
 - Class
 - gdcm::MediaStorage, [800](#)
- gdcm::UIDs, [1304](#)
- GREEN
 - gdcm::LookupTable, [779](#)
- green
 - gdcm::terminal, [110](#)
- GroupDict
 - gdcm::GroupDict, [619](#)
- GroupStringVector
 - gdcm::GroupDict, [619](#)
- GuessFromModality
 - gdcm::MediaStorage, [803](#)
- HandleBulkData
 - gdcm::XMLPrinter, [1569](#)
- HandleDataSet
 - gdcm::network::ULBasicCallback, [1372](#)
 - gdcm::network::ULConnectionCallback, [1379](#)
 - gdcm::network::ULWritingCallback, [1394](#)
- HandleDescription
 - gdcm::XMLDictReader, [1566](#)
 - gdcm::XMLPrivateDictReader, [1573](#)
- HandleEntry
 - gdcm::XMLDictReader, [1566](#)
 - gdcm::XMLPrivateDictReader, [1573](#)
- HandleEvent
 - gdcm::network::ULTransitionTable, [1392](#)
- HandleIOD
 - gdcm::TableReader, [1242](#)
- HandleIODEntry
 - gdcm::TableReader, [1242](#)
- HandleMacro
 - gdcm::TableReader, [1242](#)
- HandleMacroEntry
 - gdcm::TableReader, [1242](#)
- HandleMacroEntryDescription
 - gdcm::TableReader, [1242](#)
- HandleModule
 - gdcm::TableReader, [1242](#)
- HandleModuleEntry
 - gdcm::TableReader, [1243](#)
- HandleModuleEntryDescription
 - gdcm::TableReader, [1243](#)
- HandleModuleInclude
 - gdcm::TableReader, [1243](#)
- HandleResponse
 - gdcm::network::ULBasicCallback, [1372](#)
 - gdcm::network::ULConnectionCallback, [1379](#)
 - gdcm::network::ULWritingCallback, [1394](#)
- HangingProtocolInformationModelFIND
 - gdcm::UIDs, [1307](#)
- HangingProtocolInformationModelGET
 - gdcm::UIDs, [1311](#)
- HangingProtocolInformationModelMOVE
 - gdcm::UIDs, [1307](#)

- HangingProtocolStorage
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1307](#)
- HardcopyColorImageStorage
 - gdcm::MediaStorage, [801](#)
- HardcopyColorImageStorageSOPClassRetired
 - gdcm::UIDs, [1304](#)
- HardcopyGrayscaleImageStorage
 - gdcm::MediaStorage, [800](#)
- HardcopyGrayscaleImageStorageSOPClassRetired
 - gdcm::UIDs, [1303](#)
- HasObserver
 - gdcm::Subject, [1200](#)
- HemodynamicWaveformStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- HEVCH_265Main10ProfileLevel5__1
 - gdcm::UIDs, [1309](#)
- HEVCH_265MainProfileLevel5__1
 - gdcm::UIDs, [1309](#)
- hidden
 - gdcm::terminal, [110](#)
- HITACHI
 - gdcm::EquipmentManufacturer, [531](#)
- HotIronColorPaletteSOPInstance
 - gdcm::UIDs, [1309](#)
- HotMetalBlueColorPaletteSOPInstance
 - gdcm::UIDs, [1308](#)
- HSV
 - gdcm::PhotometricInterpretation, [928](#)
- HTJ2K
 - gdcm::TransferSyntax, [1272](#)
- HTJ2KLossless
 - gdcm::TransferSyntax, [1272](#)
- HTJ2KRPCLLossless
 - gdcm::TransferSyntax, [1272](#)
- ICBM452T1FrameofReference
 - gdcm::UIDs, [1302](#)
- ICBMSingleSubjectMRIFrameofReference
 - gdcm::UIDs, [1302](#)
- ICD11
 - gdcm::UIDs, [1308](#)
- Icon
 - gdcm::Pixmap, [945](#)
- IconDataScalarType
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1471](#)
- IconImage
 - gdcm, [87](#)
- IconImageDataExtent
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1472](#)
- IconImageFilter
 - gdcm::IconImageFilter, [622](#)
- IconImageGenerator
 - gdcm::IconImageGenerator, [625](#)
- IconNumberOfScalarComponents
 - vtkGDCMImageReader, [1457](#)
 - vtkGDCMImageReader2, [1472](#)
- ID
 - gdcm::PresentationContext, [970](#)
- ignore_char
 - gdcm::ignore_char, [628](#)
- Image
 - gdcm::Image, [633](#)
- ImageActor
 - vtkImageColorViewer, [1521](#)
- ImageApplyLookupTable
 - gdcm::ImageApplyLookupTable, [639](#)
- ImageBiomarkerStandardisationInitiative
 - gdcm::UIDs, [1308](#)
- ImageChangePhotometricInterpretation
 - gdcm::ImageChangePhotometricInterpretation, [643](#)
 - gdcm::ImageCodec, [666](#)
- ImageChangePlanarConfiguration
 - gdcm::ImageChangePlanarConfiguration, [648](#)
- ImageChangeTransferSyntax
 - gdcm::Bitmap, [261](#)
 - gdcm::ImageChangeTransferSyntax, [653](#)
- ImageCodec
 - gdcm::ImageCodec, [659](#)
- ImageConverter
 - gdcm::ImageConverter, [669](#)
- ImageFormat
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMImageReader2, [1472](#)
- ImageFragmentSplitter
 - gdcm::ImageFragmentSplitter, [672](#)
- ImageNumberOrdering
 - gdcm::SerieHelper, [1112](#)
- ImageOrientationPatient
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMImageReader2, [1472](#)
- ImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, [1303](#)
- ImagePositionPatient
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMImageReader2, [1472](#)
- ImagePositionPatientOrdering
 - gdcm::SerieHelper, [1112](#)
- ImageReader
 - gdcm::ImageReader, [683](#)
- ImageRegionReader
 - gdcm::ImageRegionReader, [688](#)
 - gdcm::JPEG2000Codec, [745](#)
 - gdcm::JPEGCodec, [759](#)

- gdcmm::JPEGLSCodec, 767
- gdcmm::RLECodec, 1047
- ImageToImageFilter
 - gdcmm::ImageToImageFilter, 692
- ImageWriter
 - gdcmm::ImageWriter, 696
- ImplantAssemblyTemplateInformationModelFIND
 - gdcmm::UIDs, 1311
- ImplantAssemblyTemplateInformationModelGET
 - gdcmm::UIDs, 1311
- ImplantAssemblyTemplateInformationModelMOVE
 - gdcmm::UIDs, 1311
- ImplantAssemblyTemplateStorage
 - gdcmm::UIDs, 1311
- ImplantationPlanSRStorage
 - gdcmm::UIDs, 1310
- ImplantTemplateGroupInformationModelFIND
 - gdcmm::UIDs, 1311
- ImplantTemplateGroupInformationModelGET
 - gdcmm::UIDs, 1311
- ImplantTemplateGroupInformationModelMOVE
 - gdcmm::UIDs, 1311
- ImplantTemplateGroupStorage
 - gdcmm::UIDs, 1311
- ImplementationClassUIDSub
 - gdcmm::network::ImplementationClassUIDSub, 698
- ImplementationUIDSub
 - gdcmm::network::ImplementationUIDSub, 699
- ImplementationVersionNameSub
 - gdcmm::network::ImplementationVersionNameSub, 700
- Implicit
 - gdcmm::TransferSyntax, 1271
- ImplicitVRBigEndianACRNEMA
 - gdcmm::TransferSyntax, 1272
- ImplicitVRBigEndianPrivateGE
 - gdcmm::TransferSyntax, 1271
- ImplicitVRLittleEndian
 - gdcmm::TransferSyntax, 1271
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
 - gdcmm::UIDs, 1301
- IncompleteLUT
 - gdcmm::LookupTable, 784
- InformationObjectDefinition Directory Reference, 61
- InitFromRQ
 - gdcmm::network::AAssociateACPDU, 118
- Initialize
 - gdcmm::network::ULConnectionInfo, 1381
- InitializeBlueLUT
 - gdcmm::LookupTable, 782
- InitializeConnection
 - gdcmm::network::ULConnection, 1376
- gdcmm::ServiceClassUser, 1119
- Initialized
 - gdcmm::LookupTable, 782
- InitializeDataSet
 - gdcmm::BaseRootQuery, 239
 - gdcmm::FindPatientRootQuery, 605
 - gdcmm::FindStudyRootQuery, 609
 - gdcmm::MovePatientRootQuery, 839
 - gdcmm::MoveStudyRootQuery, 843
 - gdcmm::WLMFindQuery, 1557
- InitializeGreenLUT
 - gdcmm::LookupTable, 782
- InitializeIncomingConnection
 - gdcmm::network::ULConnection, 1376
- InitializeLUT
 - gdcmm::LookupTable, 782
- InitializeRedLUT
 - gdcmm::LookupTable, 782
- InitializeRTStructSet
 - vtkGDCMPolyDataWriter, 1492
- InitOpenSSL
 - gdcmm::OpenSSLCryptoFactory, 877
- Input
 - gdcmm::BitmapToBitmapFilter, 265
- Insert
 - gdcmm::CommandDataSet, 323
 - gdcmm::DataSet, 395
 - gdcmm::FileMetaInformation, 578
 - gdcmm::GroupDict, 620
- InsertDataElement
 - gdcmm::DataSet, 395
 - gdcmm::Item, 725
- InsertEntry
 - gdcmm::Table, 1238
- InstallPipeline
 - vtkImageColorViewer, 1515
- InstanceAvailabilityNotificationSOPClass
 - gdcmm::UIDs, 1306
- INT12
 - gdcmm::PixelFormat, 932
- INT16
 - gdcmm::PixelFormat, 932
- INT32
 - gdcmm::PixelFormat, 932
- INT64
 - gdcmm::PixelFormat, 933
- INT8
 - gdcmm::PixelFormat, 932
- IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN
 - gdcmm::UIDs, 1308
- Interactor
 - vtkImageColorViewer, 1522
- InteractorStyle

- vtkImageColorViewer, [1522](#)
- INTERFILE
 - gdcm::CSAHeader, [351](#)
- Internal
 - gdcm::ApplicationEntity, [149](#)
 - gdcm::Attribute< Group, Element, TVR, TVM >, [167](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [175](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, [181](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, [186](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, [201](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, [206](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, [213](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, [219](#)
 - gdcm::Element< TVR, TVM >, [462](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [467](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [480](#)
 - gdcm::Element< TVR, VM::VM2_n >, [485](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [492](#)
 - gdcm::Element< TVR, VM::VM3_4 >, [497](#)
 - gdcm::Element< TVR, VM::VM3_n >, [503](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [507](#)
 - gdcm::Element< VR::OB, VM::VM1 >, [512](#)
 - gdcm::Element< VR::OW, VM::VM1 >, [517](#)
 - gdcm::LookupTable, [784](#)
 - gdcm::UI, [1282](#)
- InternalCode
 - gdcm::Coder, [311](#)
 - gdcm::JPEG12Codec, [732](#)
 - gdcm::JPEG16Codec, [737](#)
 - gdcm::JPEG8Codec, [750](#)
- Internals
 - vtkRTStructSetProperties, [1552](#)
- IntraocularLensCalculationsStorage
 - gdcm::UIDs, [1310](#)
- IntravascularOpticalCoherenceTomographyImageStorageForPresentation
 - gdcm::UIDs, [1309](#)
- IntravascularOpticalCoherenceTomographyImageStorageForProcessing
 - gdcm::UIDs, [1309](#)
- INVALID
 - gdcm::VR, [1432](#)
- Invalid
 - gdcm::Overlay, [892](#)
 - gdcm::Usage, [1405](#)
- InverseRescale
 - gdcm::Rescaler, [1038](#)
- InverseRescaleFunctionIntoBestFit
 - gdcm::Rescaler, [1038](#)
- InvokeEvent
 - gdcm::Subject, [1200](#)
- IOD
 - gdcm::IOD, [707](#)
- IODEntry
 - gdcm::IODEntry, [710](#)
- IODMapType
 - gdcm::IODs, [712](#)
- IODMapTypeConstIterator
 - gdcm::IODs, [712](#)
- IODName
 - gdcm::IODs, [712](#)
- IODs
 - gdcm::IODs, [713](#)
- IPPSorter
 - gdcm::IPPSorter, [717](#)
- IS
 - gdcm::VR, [1432](#)
- IsAETitleValid
 - gdcm::network::AAssociateRQPDU, [125](#)
- IsASCII
 - gdcm::VR, [1435](#)
- IsASCII2
 - gdcm::VR, [1435](#)
- IsBinary
 - gdcm::VR, [1435](#)
- IsBinary2
 - gdcm::VR, [1435](#)
- IsCompatible
 - gdcm::PixelFormat, [935](#)
- IsDual
 - gdcm::VR, [1436](#)
- IsEmpty
 - gdcm::Bitmap, [256](#)
 - gdcm::ByteValue, [280](#)
 - gdcm::CSAElement, [345](#)
 - gdcm::CSAHeaderDict, [356](#)
 - gdcm::Curve, [367](#)
 - gdcm::DataElement, [377](#)
 - gdcm::DataSet, [396](#)
 - gdcm::Defs, [410](#)
 - gdcm::Dict, [423](#)
 - gdcm::Dicts, [438](#)
 - gdcm::Filename, [583](#)
 - gdcm::Macros, [791](#)
 - gdcm::Modules, [835](#)
 - gdcm::Overlay, [895](#)
 - gdcm::Preamble, [965](#)
 - gdcm::PrivateDict, [989](#)
 - gdcm::SegmentHelper::BasicCodedEntry, [243](#)
 - gdcm::SequenceOfItems, [1105](#)

- IsEncapsulated
 - gdcm::TransferSyntax, [1273](#)
- IsEncoded
 - gdcm::TransferSyntax, [1274](#)
- IsExplicit
 - gdcm::TransferSyntax, [1274](#)
- IsFrameEncoder
 - gdcm::ImageCodec, [663](#)
 - gdcm::JPEG2000Codec, [744](#)
 - gdcm::JPEGCodec, [758](#)
 - gdcm::JPEGLSCodec, [766](#)
 - gdcm::RLECodec, [1046](#)
- IsGroupLength
 - gdcm::Tag, [1249](#)
- IsGroupXX
 - gdcm::Tag, [1249](#)
- IsIdentical
 - gdcm::Filename, [583](#)
- IsIllegal
 - gdcm::Tag, [1250](#)
- IsImage
 - gdcm::MediaStorage, [803](#)
- IsImplicit
 - gdcm::TransferSyntax, [1274](#)
- IsInPixelData
 - gdcm::Overlay, [895](#)
- IsKey
 - gdcm::Scanner, [1056](#)
 - gdcm::Scanner2, [1067](#)
 - gdcm::StrictScanner, [1176](#)
 - gdcm::StrictScanner2, [1187](#)
- IsLastFragment
 - gdcm::network::AAabortPDU, [114](#)
 - gdcm::network::AAAssociateACPDU, [118](#)
 - gdcm::network::AAAssociateRJPDU, [121](#)
 - gdcm::network::AAAssociateRQPDU, [126](#)
 - gdcm::network::AReleaseRPPDU, [150](#)
 - gdcm::network::AReleaseRQPDU, [153](#)
 - gdcm::network::BasePDU, [230](#)
 - gdcm::network::PDataTFPDU, [907](#)
- IsLossless
 - gdcm::PhotometricInterpretation, [929](#)
 - gdcm::TransferSyntax, [1274](#)
- IsLossy
 - gdcm::Bitmap, [256](#)
 - gdcm::ImageCodec, [664](#)
 - gdcm::PhotometricInterpretation, [929](#)
 - gdcm::TransferSyntax, [1274](#)
- IsOdd
 - gdcm::VL, [1422](#)
- IsPresentationContextAccepted
 - gdcm::ServiceClassUser, [1120](#)
- IsPrintable
 - gdcm::ByteValue, [280](#)
- IsPrivate
 - gdcm::Tag, [1250](#)
- IsPrivateCreator
 - gdcm::Tag, [1250](#)
- IsPublic
 - gdcm::Tag, [1250](#)
- IsRetired
 - gdcm::PhotometricInterpretation, [929](#)
- IsRGB8
 - gdcm::LookupTable, [782](#)
- IsRowEncoder
 - gdcm::ImageCodec, [664](#)
 - gdcm::JPEG2000Codec, [744](#)
 - gdcm::JPEGCodec, [758](#)
 - gdcm::JPEGLSCodec, [766](#)
 - gdcm::RLECodec, [1046](#)
- IsSameColorSpace
 - gdcm::PhotometricInterpretation, [929](#)
- IsStateSuspension
 - gdcm::JPEG12Codec, [732](#)
 - gdcm::JPEG16Codec, [737](#)
 - gdcm::JPEG8Codec, [750](#)
 - gdcm::JPEGCodec, [758](#)
- IsSwap
 - gdcm::VR, [1436](#)
- IsTransferSyntaxCompatible
 - gdcm::Bitmap, [256](#)
- IsUndefined
 - gdcm::MediaStorage, [804](#)
 - gdcm::VL, [1422](#)
- IsUndefinedLength
 - gdcm::DataElement, [377](#)
 - gdcm::SequenceOfItems, [1105](#)
- IsUnique
 - gdcm::DictEntry, [430](#)
- IsValid
 - gdcm::ApplicationEntity, [148](#)
 - gdcm::BoxRegion, [269](#)
 - gdcm::CodeString, [315](#)
 - gdcm::DirectionCosines, [443](#)
 - gdcm::DPath, [451](#)
 - gdcm::FileMetaInformation, [578](#)
 - gdcm::ImageCodec, [664](#)
 - gdcm::JPEGCodec, [758](#)
 - gdcm::LO, [776](#)
 - gdcm::PixelFormat, [936](#)
 - gdcm::Preamble, [965](#)
 - gdcm::Region, [1035](#)
 - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1192](#)
 - gdcm::TagPath, [1257](#)
 - gdcm::TransferSyntax, [1274](#)
 - gdcm::UIDGenerator, [1284](#)
 - gdcm::UIDGenerator, [1411](#)

- gdcmm::VM, [1429](#)
- gdcmm::VR, [1436](#)
- IsVRFile
 - gdcmm::VR, [1436](#)
- IsZero
 - gdcmm::Overlay, [895](#)
- Item
 - gdcmm::Item, [723](#)
- Items
 - gdcmm::SequenceOfItems, [1108](#)
- ItemVector
 - gdcmm::SequenceOfItems, [1102](#)
- Iterator
 - gdcmm::CSAHeaderDict, [355](#)
 - gdcmm::DataSet, [391](#)
 - gdcmm::Dict, [420](#)
 - gdcmm::SequenceOfFragments, [1094](#)
 - gdcmm::SequenceOfItems, [1102](#)
- iterator
 - gdcmm::CodeString, [314](#)
 - gdcmm::LO, [775](#)
 - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [1191](#)
- ItFileSetHt
 - gdcmm::SerieHelper, [1113](#)
- IVOCTForPresentation
 - gdcmm::MediaStorage, [801](#)
- IVOCTForProcessing
 - gdcmm::MediaStorage, [801](#)
- Join
 - gdcmm::Filename, [583](#)
- JPEG12Codec
 - gdcmm::JPEG12Codec, [731](#)
- JPEG16Codec
 - gdcmm::JPEG16Codec, [736](#)
- JPEG2000
 - gdcmm::TransferSyntax, [1272](#)
- JPEG2000_COMPRESSION
 - vtkGDCMImageWriter, [1476](#)
- JPEG2000Codec
 - gdcmm::JPEG2000Codec, [741](#)
- JPEG2000ImageCompression
 - gdcmm::UIDs, [1302](#)
- JPEG2000ImageCompressionLosslessOnly
 - gdcmm::UIDs, [1301](#)
- JPEG2000Lossless
 - gdcmm::TransferSyntax, [1272](#)
- JPEG2000Part2
 - gdcmm::TransferSyntax, [1272](#)
- JPEG2000Part2Lossless
 - gdcmm::TransferSyntax, [1272](#)
- JPEG2000Part2MulticomponentImageCompression
 - gdcmm::UIDs, [1302](#)
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly
 - gdcmm::UIDs, [1302](#)
- JPEG8Codec
 - gdcmm::JPEG8Codec, [750](#)
- JPEG_COMPRESSION
 - vtkGDCMImageWriter, [1476](#)
- JPEGBaselineProcess1
 - gdcmm::TransferSyntax, [1271](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression
 - gdcmm::UIDs, [1301](#)
- JPEGCodec
 - gdcmm::JPEGCodec, [755](#)
- JPEGExtendedHierarchicalProcess1618Retired
 - gdcmm::UIDs, [1301](#)
- JPEGExtendedHierarchicalProcess1719Retired
 - gdcmm::UIDs, [1301](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only
 - gdcmm::UIDs, [1301](#)
- JPEGExtendedProcess2_4
 - gdcmm::TransferSyntax, [1271](#)
- JPEGExtendedProcess35Retired
 - gdcmm::UIDs, [1301](#)
- JPEGExtendedProcess3_5
 - gdcmm::TransferSyntax, [1271](#)
- JPEGFullProgressionHierarchicalProcess2426Retired
 - gdcmm::UIDs, [1301](#)
- JPEGFullProgressionHierarchicalProcess2527Retired
 - gdcmm::UIDs, [1301](#)
- JPEGFullProgressionNonHierarchicalProcess1012Retired
 - gdcmm::UIDs, [1301](#)
- JPEGFullProgressionNonHierarchicalProcess1113Retired
 - gdcmm::UIDs, [1301](#)
- JPEGFullProgressionProcess10_12
 - gdcmm::TransferSyntax, [1271](#)
- JPEGLosslessHierarchicalProcess28Retired
 - gdcmm::UIDs, [1301](#)
- JPEGLosslessHierarchicalProcess29Retired
 - gdcmm::UIDs, [1301](#)
- JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression
 - gdcmm::UIDs, [1301](#)
- JPEGLosslessNonHierarchicalProcess14
 - gdcmm::UIDs, [1301](#)
- JPEGLosslessNonHierarchicalProcess15Retired
 - gdcmm::UIDs, [1301](#)

- JPEGLosslessProcess14
 - gdcm::TransferSyntax, [1271](#)
- JPEGLosslessProcess14_1
 - gdcm::TransferSyntax, [1272](#)
- JPEGLS_COMPRESSION
 - vtkGDCMImageWriter, [1476](#)
- JPEGLSCodec
 - gdcm::JPEGLSCodec, [763](#)
- JPEGLSLossless
 - gdcm::TransferSyntax, [1272](#)
- JPEGLSLosslessImageCompression
 - gdcm::UIDs, [1301](#)
- JPEGLSLossyNearLosslessImageCompression
 - gdcm::UIDs, [1301](#)
- JPEGLSNearLossless
 - gdcm::TransferSyntax, [1272](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired
 - gdcm::UIDs, [1301](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired
 - gdcm::UIDs, [1301](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
 - gdcm::UIDs, [1301](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
 - gdcm::UIDs, [1301](#)
- JPEGSpectralSelectionProcess6_8
 - gdcm::TransferSyntax, [1271](#)
- JPIPReferenced
 - gdcm::TransferSyntax, [1272](#)
 - gdcm::UIDs, [1302](#)
- JPIPReferencedDeflate
 - gdcm::UIDs, [1302](#)
- JSON
 - gdcm::JSON, [768](#)
- JunkAfterDocElementError
 - gdcm::Parser, [902](#)
- KAKADUCodec
 - gdcm::KAKADUCodec, [772](#)
- KeratometryMeasurementsStorage
 - gdcm::UIDs, [1309](#)
- KeyField
 - gdcm::CSAElement, [347](#)
- KeyObjectSelectionDocument
 - gdcm::MediaStorage, [800](#)
- KeyObjectSelectionDocumentStorage
 - gdcm::UIDs, [1305](#)
- KeyValuePairArrayType
 - gdcm::CompositeNetworkFunctions, [326](#)
- KeyValuePairType
 - gdcm::CompositeNetworkFunctions, [326](#)
- KODAK
 - gdcm::EquipmentManufacturer, [531](#)
- LD_ALL
 - gdcm, [91](#)
- LD_NOSEQ
 - gdcm, [91](#)
- LD_NOSHADOW
 - gdcm, [91](#)
- LD_NOSHADOWSEQ
 - gdcm, [91](#)
- LeadECGWaveformStorage
 - gdcm::MediaStorage, [799](#)
- LegacyConvertedEnhancedCTImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- LegacyConvertedEnhancedMRImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- LegacyConvertedEnhancedPETImageStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- LensometryMeasurementsStorage
 - gdcm::UIDs, [1309](#)
- Level
 - vtkImageMapToWindowLevelColors2, [1532](#)
- LINE
 - gdcm::MeshPrimitive, [815](#)
- ListCharSets
 - gdcm::QueryFactory, [1008](#)
- LittleEndian
 - gdcm::SwapCode, [1227](#)
- LO
 - gdcm::LO, [776](#)
 - gdcm::VR, [1432](#)
- Load
 - gdcm::Directory, [446](#)
 - gdcm::MrProtocol, [845](#)
- LOADBULKDATA
 - gdcm::XMLPrinter, [1568](#)
- LoadDefault
 - gdcm::CSAHeaderDict, [356](#)
 - gdcm::Dict, [423](#)
 - gdcm::PrivateDict, [989](#)
- LoadDefaults
 - gdcm::Defs, [410](#)
 - gdcm::Dicts, [438](#)
- LoadFromDataElement
 - gdcm::CSAHeader, [353](#)
 - gdcm::PDBHeader, [913](#)
- LoadFromFile
 - gdcm::Defs, [410](#)
- LoadIconImage
 - vtkGDCMImageReader, [1458](#)

- vtkGDCMImageReader2, [1472](#)
- LoadImageFromFiles
 - gdcmm::DirectoryHelper, [450](#)
- LoadOverlays
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMImageReader2, [1472](#)
- LoadResourcesFiles
 - gdcmm::Global, [617](#)
- LoadSingleFile
 - vtkGDCMImageReader, [1449](#)
 - vtkGDCMImageReader2, [1463](#)
- Locate
 - gdcmm::Global, [617](#)
- LOComp
 - gdcmm, [87](#)
- LodModeType
 - gdcmm, [90](#)
- LookupTable
 - gdcmm::LookupTable, [779](#), [780](#)
 - vtkImageMapToColors16, [1528](#)
- LookupTableType
 - gdcmm::LookupTable, [779](#)
- LossyFlag
 - gdcmm::Bitmap, [261](#)
 - gdcmm::ImageCodec, [667](#)
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMImageReader2, [1472](#)
- LT
 - gdcmm::VR, [1432](#)
- LTCComp
 - gdcmm, [87](#)
- LUT
 - gdcmm::Bitmap, [261](#)
 - gdcmm::ImageCodec, [667](#)
- LUTPtr
 - gdcmm::Bitmap, [251](#)
 - gdcmm::ImageCodec, [659](#)
- m_char
 - gdcmm::ignore_char, [628](#)
- m_ConstMemberFunction
 - gdcmm::MemberCommand< T >, [811](#)
- m_DataSet
 - gdcmm::DataSetEvent, [403](#)
- m_MemberFunction
 - gdcmm::MemberCommand< T >, [811](#)
 - gdcmm::SimpleMemberCommand< T >, [1131](#)
- m_This
 - gdcmm::MemberCommand< T >, [811](#)
 - gdcmm::SimpleMemberCommand< T >, [1131](#)
- Macro
 - gdcmm::Macro, [788](#)
- MacroEntry
 - gdcmm, [87](#)
- Macros
 - gdcmm::Macros, [790](#)
- mAction
 - gdcmm::network::Transition, [1278](#)
- MacularGridThicknessandVolumeReportStorage
 - gdcmm::UIDs, [1310](#)
- magenta
 - gdcmm::terminal, [110](#)
- MAGNIFIED
 - gdcmm::Spacing, [1150](#)
- MakeDirectory
 - gdcmm::System, [1234](#)
- MakeNew
 - gdcmm::network::Transition, [1278](#)
- MakeObject
 - gdcmm::AnonymizeEvent, [134](#)
 - gdcmm::DataEvent, [387](#)
 - gdcmm::DataSetEvent, [403](#)
 - gdcmm::Event, [534](#)
 - gdcmm::FileNameEvent, [587](#)
 - gdcmm::ProgressEvent, [998](#)
- MammographyCADSR
 - gdcmm::MediaStorage, [800](#)
- MammographyCADSRStorage
 - gdcmm::UIDs, [1305](#)
- Mandatory
 - gdcmm::Usage, [1405](#)
- MANUAL
 - gdcmm::Segment, [1072](#)
- MapCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, [355](#)
- MapDictEntry
 - gdcmm::Dict, [421](#)
- MapIODEntry
 - gdcmm::IOD, [707](#)
- MapModuleEntry
 - gdcmm::Macro, [787](#)
 - gdcmm::Module, [827](#)
- MappingType
 - gdcmm::Scanner, [1052](#)
 - gdcmm::StrictScanner, [1172](#)
- MapScalarsThroughTable2
 - vtkLookupTable16, [1541](#)
- MapTableEntry
 - gdcmm::Table, [1238](#)
- MARCONI
 - gdcmm::EquipmentManufacturer, [531](#)
- Match
 - gdcmm::DPath, [451](#)
- MaximumLengthSub
 - gdcmm::network::MaximumLengthSub, [792](#)
- MaxLength
 - gdcmm::ApplicationEntity, [149](#)
 - gdcmm::PersonName, [921](#)

- MaxNumberOfComponents
 - gdcm::ApplicationEntity, [149](#)
 - gdcm::PersonName, [921](#)
- MaxPrintLength
 - gdcm::Printer, [987](#)
- MayoClinicNonradiologicalImagesSBSSAnatomical-
SurfaceRegionGuide
 - gdcm::UIDs, [1308](#)
- mConnection
 - gdcm::network::ULConnectionManager, [1389](#)
- MD5DataImagesType
 - gdcm::Testing, [1259](#)
- MD5MetaImagesType
 - vtkGDCMTesting, [1496](#)
- mDataSet
 - gdcm::BaseQuery, [236](#)
- MediaCreationManagementSOPClassUID
 - gdcm::UIDs, [1304](#)
- MediaStorage
 - gdcm::MediaStorage, [802](#)
- MediaStorageAndFileFormat Directory Reference, [62](#)
- MediaStorageDataFileType
 - gdcm::Testing, [1259](#)
- MediaStorageDirectoryStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1302](#)
- MedicalImageProperties
 - vtkGDCMImageReader, [1458](#)
 - vtkGDCMPolyDataReader, [1489](#)
 - vtkGDCMPolyDataWriter, [1494](#)
- mElementOffsets
 - gdcm::StreamImageWriter, [1167](#)
- mElementOffsets1
 - gdcm::StreamImageWriter, [1167](#)
- MemberCommand
 - gdcm::MemberCommand< T >, [809](#)
- mEnd
 - gdcm::network::Transition, [1278](#)
- MeshPrimitive
 - gdcm::MeshPrimitive, [815](#)
- MessageExchangeDefinition Directory Reference, [65](#)
- MessageID
 - gdcm::network::CEchoRQ, [291](#)
- MetaInformationTS
 - gdcm::FileMetaInformation, [581](#)
- mHelpDescription
 - gdcm::BaseRootQuery, [240](#)
- mImage
 - gdcm::BaseRootQuery, [240](#)
- mImplicit
 - gdcm::network::ULConnectionCallback, [1380](#)
- ModalityPerformedProcedureStepCreateQuery
 - gdcm::ModalityPerformedProcedureStepCreate-
Query, [820](#)
- ModalityPerformedProcedureStepNotificationSOP-
Class
 - gdcm::UIDs, [1303](#)
- ModalityPerformedProcedureStepRetrieveSOPClass
 - gdcm::UIDs, [1303](#)
- ModalityPerformedProcedureStepSetQuery
 - gdcm::ModalityPerformedProcedureStepSet-
Query, [823](#)
- ModalityPerformedProcedureStepSOPClass
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1303](#)
- ModalityWorklistInformationModelFIND
 - gdcm::UIDs, [1306](#)
- Mode
 - gdcm::terminal, [110](#)
- Module
 - gdcm::Module, [827](#)
- ModuleEntry
 - gdcm::ModuleEntry, [831](#)
- ModuleMapType
 - gdcm::Macros, [790](#)
 - gdcm::Modules, [834](#)
- Modules
 - gdcm::Modules, [834](#)
- MONOCHROME1
 - gdcm::PhotometricInterpretation, [927](#)
- MONOCHROME2
 - gdcm::PhotometricInterpretation, [927](#)
- MouseGenomeInitiativeMGI
 - gdcm::UIDs, [1308](#)
- MovePatientRootQuery
 - gdcm::MovePatientRootQuery, [838](#)
- MoveStudyRootQuery
 - gdcm::MoveStudyRootQuery, [842](#)
- mPatient
 - gdcm::BaseRootQuery, [240](#)
- MPEG2MainProfile
 - gdcm::TransferSyntax, [1272](#)
- MPEG2MainProfileHighLevel
 - gdcm::TransferSyntax, [1272](#)
 - gdcm::UIDs, [1308](#)
- MPEG2MainProfileMainLevel
 - gdcm::UIDs, [1302](#)
- MPEG4AVCH264BDcompatibleHighProfileLevel4_1
 - gdcm::TransferSyntax, [1272](#)
- MPEG4AVCH264HighProfileLevel4_1
 - gdcm::TransferSyntax, [1272](#)
- MPEG4AVCH_264BDcompatibleHighProfileLevel4_1
 - gdcm::UIDs, [1308](#)
- MPEG4AVCH_264HighProfileLevel4_1
 - gdcm::UIDs, [1308](#)
- MPEG4AVCH_264HighProfileLevel4_2For2DVideo
 - gdcm::UIDs, [1309](#)
- MPEG4AVCH_264HighProfileLevel4_2For3DVideo

- gdcmm::UIDs, [1309](#)
- MPEG4AVCH_264StereoHighProfileLevel4_2
 - gdcmm::UIDs, [1309](#)
- MPTType
 - gdcmm::MeshPrimitive, [814](#)
- MPTType_END
 - gdcmm::MeshPrimitive, [815](#)
- MRImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- mRootType
 - gdcmm::BaseRootQuery, [240](#)
- MrProtocol
 - gdcmm::MrProtocol, [844](#)
- MRsSpectroscopyStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- MS_END
 - gdcmm::MediaStorage, [801](#)
- mSecondaryConnection
 - gdcmm::network::ULConnectionManager, [1389](#)
- mSeries
 - gdcmm::BaseRootQuery, [241](#)
- mSopInstanceUID
 - gdcmm::BaseQuery, [236](#)
- mspFile
 - gdcmm::StreamImageWriter, [1167](#)
- mStudy
 - gdcmm::BaseRootQuery, [241](#)
- MSType
 - gdcmm::MediaStorage, [799](#)
- mTransitions
 - gdcmm::network::ULConnectionManager, [1389](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- MultiframeSingleBitSecondaryCaptureImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- MultiframeTrueColorSecondaryCaptureImageStorage
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- MultipleVolumeRenderingVolumetricPresentationStateStorage
 - gdcmm::UIDs, [1309](#)
- mWriter
 - gdcmm::StreamImageWriter, [1168](#)
- mXMax
 - gdcmm::StreamImageWriter, [1168](#)
- mXMin
 - gdcmm::StreamImageWriter, [1168](#)
- mYMax
 - gdcmm::StreamImageWriter, [1168](#)
- mYMin
 - gdcmm::StreamImageWriter, [1168](#)
- mZMax
 - gdcmm::StreamImageWriter, [1168](#)
- mZMin
 - gdcmm::StreamImageWriter, [1168](#)
- N_ACTION_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_ACTION_RSP
 - gdcmm::network::DIMSE, [440](#)
- N_CREATE_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_CREATE_RSP
 - gdcmm::network::DIMSE, [440](#)
- N_DELETE_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_DELETE_RSP
 - gdcmm::network::DIMSE, [440](#)
- N_EVENT_REPORT_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_EVENT_REPORT_RSP
 - gdcmm::network::DIMSE, [440](#)
- N_GET_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_GET_RSP
 - gdcmm::network::DIMSE, [440](#)
- N_SET_RQ
 - gdcmm::network::DIMSE, [440](#)
- N_SET_RSP
 - gdcmm::network::DIMSE, [440](#)
- NAction
 - gdcmm::NormalizedNetworkFunctions, [867](#)
- Name
 - gdcmm::ModuleEntry, [833](#)
- NameField
 - gdcmm::CSAElement, [348](#)
 - gdcmm::PDSElement, [910](#)
- NativeDICOMModel
 - gdcmm::UIDs, [1311](#)
- NCreate
 - gdcmm::NormalizedNetworkFunctions, [867](#)
- NDelete
 - gdcmm::NormalizedNetworkFunctions, [868](#)
- NeedByteSwap
 - gdcmm::Bitmap, [261](#)
 - gdcmm::ImageCodec, [667](#)
- NeedOverlayCleanup
 - gdcmm::ImageCodec, [667](#)

- NegotiatedType
 - gdcmm::TransferSyntax, [1271](#)
- NestedMacroEntries
 - gdcmm, [87](#)
- NestedModuleEntries
 - gdcmm::NestedModuleEntries, [857](#)
- NEventReport
 - gdcmm::NormalizedNetworkFunctions, [868](#)
- New
 - gdcmm::Anonymizer, [141](#)
 - gdcmm::Cleaner, [301](#)
 - gdcmm::FileChangeTransferSyntax, [560](#)
 - gdcmm::FileStreamer, [597](#)
 - gdcmm::MemberCommand< T >, [810](#)
 - gdcmm::Scanner, [1056](#)
 - gdcmm::Scanner2, [1067](#)
 - gdcmm::SequenceOfFragments, [1097](#)
 - gdcmm::SequenceOfItems, [1106](#)
 - gdcmm::ServiceClassUser, [1120](#)
 - gdcmm::SimpleMemberCommand< T >, [1130](#)
 - gdcmm::StrictScanner, [1176](#)
 - gdcmm::StrictScanner2, [1187](#)
 - vtkGDCMImageReader, [1449](#)
 - vtkGDCMImageReader2, [1464](#)
 - vtkGDCMImageWriter, [1477](#)
 - vtkGDCMMedicalImageProperties, [1484](#)
 - vtkGDCMPolyDataReader, [1487](#)
 - vtkGDCMPolyDataWriter, [1492](#)
 - vtkGDCMTesting, [1497](#)
 - vtkGDCMThreadedImageReader, [1501](#)
 - vtkGDCMThreadedImageReader2, [1505](#)
 - vtkImageColorViewer, [1515](#)
 - vtkImageMapToColors16, [1525](#)
 - vtkImageMapToWindowLevelColors2, [1531](#)
 - vtkImagePlanarComponentsToComponents, [1534](#)
 - vtkImageRGBToYBR, [1536](#)
 - vtkImageYBRToRGB, [1539](#)
 - vtkLookupTable16, [1542](#)
 - vtkRTStructSetProperties, [1548](#)
- NewYorkUniversityMelanomaClinicalCooperative-Group
 - gdcmm::UIDs, [1308](#)
- NGet
 - gdcmm::NormalizedNetworkFunctions, [868](#)
- NO
 - gdcmm::Surface, [1204](#)
- NO_COMPRESSION
 - vtkGDCMImageWriter, [1476](#)
- NoElementsError
 - gdcmm::Parser, [902](#)
- NoError
 - gdcmm::Parser, [902](#)
- NOMAGIC
 - gdcmm::CSAHeader, [350](#)
- NoMemoryError
 - gdcmm::Parser, [902](#)
- NoObject
 - gdcmm::MediaStorage, [801](#)
- NoOfItemsField
 - gdcmm::CSAElement, [348](#)
- Norm
 - gdcmm::DirectionCosines, [443](#)
- Normal
 - gdcmm::MrProtocol::Slice, [1135](#)
- Normalize
 - gdcmm::DirectionCosines, [443](#)
- NSet
 - gdcmm::NormalizedNetworkFunctions, [868](#)
- NuclearMedicineImageStorage
 - gdcmm::MediaStorage, [800](#)
 - gdcmm::UIDs, [1305](#)
- NuclearMedicineImageStorageRetired
 - gdcmm::MediaStorage, [799](#)
 - gdcmm::UIDs, [1304](#)
- Null0
 - gdcmm::UIDs, [1309](#)
- Null1
 - gdcmm::UIDs, [1309](#)
- NumberOfDimensions
 - gdcmm::Bitmap, [261](#)
 - gdcmm::ImageCodec, [667](#)
- NumberOfIconImages
 - vtkGDCMImageReader, [1459](#)
 - vtkGDCMImageReader2, [1473](#)
- NumberOfOverlays
 - vtkGDCMImageReader, [1459](#)
 - vtkGDCMImageReader2, [1473](#)
- NumberOfSurfaces
 - gdcmm::SurfaceWriter, [1225](#)
- OB
 - gdcmm::VR, [1432](#)
- OB_OW
 - gdcmm::VR, [1433](#)
- Object
 - gdcmm::Object, [873](#)
- ObjectEnd
 - gdcmm::MediaStorage, [802](#)
- ObjectType
 - gdcmm::MediaStorage, [801](#)
- OBLIQUE
 - gdcmm::Orientation, [887](#)
- OD
 - gdcmm::VR, [1432](#)
- OF
 - gdcmm::VR, [1432](#)
- Ofstream

- gdcmm::Writer, [1564](#)
- OL
 - gdcmm::VR, [1433](#)
- OnlyUUID
 - gdcmm::XMLPrinter, [1568](#)
- OPENSSL
 - gdcmm::CryptoFactory, [336](#)
- OpenSSLCryptoFactory
 - gdcmm::OpenSSLCryptoFactory, [876](#)
- OpenSSLCryptographicMessageSyntax
 - gdcmm::OpenSSLCryptographicMessageSyntax, [879](#)
- OPENSSLP7
 - gdcmm::CryptoFactory, [336](#)
- OpenSSLP7CryptoFactory
 - gdcmm::OpenSSLP7CryptoFactory, [882](#)
- OpenSSLP7CryptographicMessageSyntax
 - gdcmm::OpenSSLP7CryptographicMessageSyntax, [884](#)
- operator const char *
 - gdcmm::ConstCharWrapper, [331](#)
 - gdcmm::Filename, [583](#)
 - gdcmm::String< TDelimiter, TMaxLength, TPad-Char >, [1192](#)
- operator const double *
 - gdcmm::DirectionCosines, [443](#)
- operator const std::vector< char > &
 - gdcmm::ByteValue, [280](#)
- operator MStype
 - gdcmm::MediaStorage, [804](#)
- operator ObjectType *
 - gdcmm::SmartPointer< ObjectType >, [1139](#)
- operator PType
 - gdcmm::PhotometricInterpretation, [929](#)
- operator ScalarType
 - gdcmm::PixelFormat, [936](#)
- operator SwapCode::SwapCodeType
 - gdcmm::SwapCode, [1227](#)
- operator TStype
 - gdcmm::TransferSyntax, [1274](#)
 - gdcmm::UIDs, [1328](#)
- operator TypeType
 - gdcmm::Type, [1281](#)
- operator uint32_t
 - gdcmm::VL, [1423](#)
- operator UsageType
 - gdcmm::Usage, [1406](#)
- operator VMType
 - gdcmm::VM, [1429](#)
- operator VRType
 - gdcmm::VR, [1436](#)
- operator!=
 - gdcmm, [92](#)
- gdcmm::Attribute< Group, Element, TVR, TVM >, [164](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [191](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [211](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
- gdcmm::CodeString, [316](#)
- gdcmm::PixelFormat, [936](#)
- gdcmm::PrivateTag, [993](#)
- gdcmm::Tag, [1251](#)
- operator<
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [164](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [191](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
 - gdcmm::CSAElement, [345](#)
 - gdcmm::CSAHeaderDictEntry, [359](#)
 - gdcmm::DataElement, [377](#)
 - gdcmm::DPath, [452](#)
 - gdcmm::PrivateTag, [994](#)
 - gdcmm::Tag, [1251](#)
- operator<<
 - gdcmm, [92–101](#)
 - gdcmm::BasicOffsetTable, [248](#)
 - gdcmm::CodeString, [316](#)
 - gdcmm::CommandDataSet, [324](#)
 - gdcmm::CSAElement, [347](#)
 - gdcmm::CSAHeader, [354](#)

- gdcm::CSAHeaderDict, [357](#)
- gdcm::CSAHeaderDictEntry, [360](#)
- gdcm::DataElement, [382](#)
- gdcm::DataSet, [400](#)
- gdcm::Dict, [423](#)
- gdcm::DictEntry, [431](#)
- gdcm::Dicts, [438](#)
- gdcm::Directory, [447](#)
- gdcm::DPath, [452](#)
- gdcm::File, [552](#)
- gdcm::FileMetaInformation, [580](#)
- gdcm::FileSet, [593](#)
- gdcm::Fragment, [614](#)
- gdcm::Global, [618](#)
- gdcm::GroupDict, [621](#)
- gdcm::IOD, [708](#)
- gdcm::IODEntry, [711](#)
- gdcm::IODs, [714](#)
- gdcm::Item, [726](#)
- gdcm::Macro, [789](#)
- gdcm::Macros, [791](#)
- gdcm::MediaStorage, [805](#)
- gdcm::Module, [829](#)
- gdcm::ModuleEntry, [833](#)
- gdcm::Modules, [835](#)
- gdcm::MrProtocol, [846](#)
- gdcm::NestedModuleEntries, [858](#)
- gdcm::Object, [874](#)
- gdcm::Orientation, [889](#)
- gdcm::PDBelement, [910](#)
- gdcm::PDBHeader, [913](#)
- gdcm::PhotometricInterpretation, [930](#)
- gdcm::PixelFormat, [938](#)
- gdcm::Preamble, [966](#)
- gdcm::PrivateDict, [989](#)
- gdcm::PrivateTag, [995](#)
- gdcm::Scanner, [1058](#)
- gdcm::Scanner2, [1069](#)
- gdcm::Sorter, [1147](#)
- gdcm::StrictScanner, [1178](#)
- gdcm::StrictScanner2, [1188](#)
- gdcm::SwapCode, [1228](#)
- gdcm::Table, [1239](#)
- gdcm::Tag, [1255](#)
- gdcm::TransferSyntax, [1275](#)
- gdcm::Type, [1281](#)
- gdcm::UI, [1282](#)
- gdcm::Usage, [1406](#)
- gdcm::Version, [1420](#)
- gdcm::VL, [1424](#)
- gdcm::VM, [1429](#)
- gdcm::VR, [1437](#)
- operator<=
 - gdcm::Tag, [1251](#)
- operator>>
 - gdcm, [101](#)
 - gdcm::Tag, [1255](#)
- operator()
 - gdcm::DataSet, [396](#)
 - gdcm::Scanner2::ltstr, [785](#)
 - gdcm::Scanner::ltstr, [785](#)
 - gdcm::StrictScanner2::ltstr, [786](#)
 - gdcm::StrictScanner::ltstr, [786](#)
- operator++
 - gdcm::VL, [1423](#)
- operator+=
 - gdcm::VL, [1423](#)
- operator->
 - gdcm::SmartPointer< ObjectType >, [1139](#)
- operator=
 - gdcm::AnonymizeEvent, [134](#)
 - gdcm::ASN1, [156](#)
 - gdcm::Base64, [224](#)
 - gdcm::BoxRegion, [269](#)
 - gdcm::ByteSwapFilter, [274](#)
 - gdcm::ByteValue, [281](#)
 - gdcm::Command, [319](#)
 - gdcm::CryptographicMessageSyntax, [339](#)
 - gdcm::CSAElement, [345](#)
 - gdcm::CSAHeaderDict, [356](#)
 - gdcm::DataElement, [378](#)
 - gdcm::DataEvent, [387](#)
 - gdcm::DataSet, [396](#)
 - gdcm::DataSetEvent, [403](#)
 - gdcm::Defs, [410](#)
 - gdcm::Dict, [423](#)
 - gdcm::Dicts, [438](#)
 - gdcm::Element< TVR, VM::VM1_n >, [471](#)
 - gdcm::Event, [534](#)
 - gdcm::FileMetaInformation, [578](#)
 - gdcm::FileNameEvent, [587](#)
 - gdcm::Global, [618](#)
 - gdcm::MemberCommand< T >, [810](#)
 - gdcm::network::ULAction, [1332](#)
 - gdcm::network::ULConnection, [1376](#)
 - gdcm::network::UserInformation, [1409](#)
 - gdcm::Object, [874](#)
 - gdcm::Overlay, [896](#)
 - gdcm::ParseException, [900](#)
 - gdcm::Preamble, [965](#)
 - gdcm::PrivateTag, [994](#)
 - gdcm::ProgressEvent, [998](#)
 - gdcm::SequenceOfItems, [1106](#)
 - gdcm::ServiceClassUser, [1120](#)
 - gdcm::SHA1, [1125](#)
 - gdcm::SimpleMemberCommand< T >, [1130](#)
 - gdcm::SimpleSubjectWatcher, [1132](#)
 - gdcm::SmartPointer< ObjectType >, [1139](#), [1140](#)

- gdcmm::Table, [1239](#)
- gdcmm::Tag, [1251](#)
- operator==
 - gdcmm, [101](#)
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [164](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [192](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
 - gdcmm::ByteValue, [281](#)
 - gdcmm::CodeString, [316](#)
 - gdcmm::CSAElement, [345](#)
 - gdcmm::DataElement, [378](#)
 - gdcmm::network::AbstractSyntax, [130](#)
 - gdcmm::network::PresentationContextRQ, [978](#)
 - gdcmm::network::TransferSyntaxSub, [1276](#)
 - gdcmm::PDSElement, [909](#)
 - gdcmm::PixelFormat, [936](#)
 - gdcmm::PresentationContext, [969](#)
 - gdcmm::PrivateTag, [994](#)
 - gdcmm::SequenceOfFragments, [1097](#)
 - gdcmm::SequenceOfItems, [1106](#)
 - gdcmm::Tag, [1251](#)
 - gdcmm::Value, [1416](#)
- operator[]
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [164](#), [165](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [179](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [192](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [199](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [217](#)
- gdcmm::DataSet, [396](#)
- gdcmm::Element< TVR, TVM >, [460](#)
- gdcmm::Element< TVR, VM::VM1_2 >, [466](#)
- gdcmm::Element< TVR, VM::VM1_n >, [471](#)
- gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
- gdcmm::Element< TVR, VM::VM2_n >, [484](#)
- gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
- gdcmm::Element< TVR, VM::VM3_4 >, [496](#)
- gdcmm::Element< TVR, VM::VM3_n >, [502](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [506](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [510](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [515](#)
- gdcmm::Tag, [1251](#), [1252](#)
- operator*
 - gdcmm::SmartPointer< ObjectType >, [1139](#)
- OphthalmicAxialMeasurementsStorage
 - gdcmm::UIDs, [1309](#)
- OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage
 - gdcmm::UIDs, [1309](#)
- OphthalmicOpticalCoherenceTomographyEnFaceImageStorage
 - gdcmm::UIDs, [1309](#)
- OphthalmicPhotography16BitImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- OphthalmicPhotography8BitImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- OphthalmicThicknessMapStorage
 - gdcmm::UIDs, [1310](#)
- OphthalmicTomographyImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- OphthalmicVisualFieldStaticPerimetryMeasurementsStorage
 - gdcmm::UIDs, [1310](#)
- OrderFileList
 - gdcmm::SerieHelper, [1112](#)
- Orientation
 - gdcmm::Orientation, [887](#)
- OrientationType
 - gdcmm::Orientation, [887](#)
- Output
 - gdcmm::BitmapToBitmapFilter, [265](#)
- OutputFormat
 - vtkImageMapToColors16, [1528](#)
- OutputTypes
 - gdcmm::DictConverter, [425](#)
- OV
 - gdcmm::VR, [1433](#)
- Overlay

- gdcmm::Overlay, [892](#), [893](#)
 - OverlayImageActor
 - vtkImageColorViewer, [1522](#)
 - Overlays
 - gdcmm::Pixmap, [946](#)
 - OverlayType
 - gdcmm::Overlay, [892](#)
 - OW
 - gdcmm::VR, [1433](#)
 - Pack
 - gdcmm::Unpacker12Bits, [1403](#)
 - Padding
 - gdcmm::ApplicationEntity, [149](#)
 - gdcmm::PersonName, [922](#)
 - PALETTE_COLOR
 - gdcmm::PhotometricInterpretation, [928](#)
 - Papyrus3ImplicitVRLittleEndian
 - gdcmm::UIDs, [1308](#)
 - ParametricMapStorage
 - gdcmm::UIDs, [1309](#)
 - Parent
 - gdcmm::Element< TVR, VM::VM1_2 >, [465](#)
 - gdcmm::Element< TVR, VM::VM2_2n >, [477](#)
 - gdcmm::Element< TVR, VM::VM2_n >, [483](#)
 - gdcmm::Element< TVR, VM::VM3_3n >, [489](#)
 - gdcmm::Element< TVR, VM::VM3_4 >, [495](#)
 - gdcmm::Element< TVR, VM::VM3_n >, [501](#)
 - Parse
 - gdcmm::Parser, [903](#)
 - ParseBuffer
 - gdcmm::Parser, [903](#)
 - ParseCertificateFile
 - gdcmm::CAPICryptographicMessageSyntax, [288](#)
 - gdcmm::CryptographicMessageSyntax, [340](#)
 - gdcmm::OpenSSLCryptographicMessageSyntax, [879](#)
 - gdcmm::OpenSSL7CryptographicMessageSyntax, [885](#)
 - ParseDateTime
 - gdcmm::System, [1235](#)
 - ParseDump
 - gdcmm::ASN1, [156](#)
 - ParseDumpFile
 - gdcmm::ASN1, [156](#)
 - ParseException
 - gdcmm::ParseException, [899](#), [900](#)
 - ParseKeyFile
 - gdcmm::CAPICryptographicMessageSyntax, [288](#)
 - gdcmm::CryptographicMessageSyntax, [340](#)
 - gdcmm::OpenSSLCryptographicMessageSyntax, [880](#)
 - gdcmm::OpenSSL7CryptographicMessageSyntax, [885](#)
 - Parser
 - gdcmm::Parser, [902](#)
 - PassAlphaToOutput
 - vtkImageMapToColors16, [1528](#)
 - Patient
 - gdcmm::Patient, [904](#)
 - PatientRadiationDoseSRStorage
 - gdcmm::UIDs, [1310](#)
 - PatientRootQueryRetrieveInformationModelFIND
 - gdcmm::UIDs, [1306](#)
 - PatientRootQueryRetrieveInformationModelGET
 - gdcmm::UIDs, [1306](#)
 - PatientRootQueryRetrieveInformationModelMOVE
 - gdcmm::UIDs, [1306](#)
 - PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
 - gdcmm::UIDs, [1306](#)
 - PatientStudyOnlyQueryRetrieveInformationModelGETRetired
 - gdcmm::UIDs, [1306](#)
 - PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
 - gdcmm::UIDs, [1306](#)
- PDataTFPDU
 - gdcmm::network::PDataTFPDU, [906](#)
 - PDBElement
 - gdcmm::PDBElement, [909](#)
 - PDBHeader
 - gdcmm::PDBHeader, [912](#)
 - PDF
 - gdcmm::MediaStorage, [802](#)
 - PDFCodec
 - gdcmm::PDFCodec, [915](#)
 - PerformAction
 - gdcmm::network::ULAction, [1332](#)
 - gdcmm::network::ULActionAA1, [1334](#)
 - gdcmm::network::ULActionAA2, [1335](#)
 - gdcmm::network::ULActionAA3, [1336](#)
 - gdcmm::network::ULActionAA4, [1338](#)
 - gdcmm::network::ULActionAA5, [1339](#)
 - gdcmm::network::ULActionAA6, [1340](#)
 - gdcmm::network::ULActionAA7, [1342](#)
 - gdcmm::network::ULActionAA8, [1343](#)
 - gdcmm::network::ULActionAE1, [1344](#)
 - gdcmm::network::ULActionAE2, [1346](#)
 - gdcmm::network::ULActionAE3, [1347](#)
 - gdcmm::network::ULActionAE4, [1348](#)
 - gdcmm::network::ULActionAE5, [1350](#)
 - gdcmm::network::ULActionAE6, [1351](#)
 - gdcmm::network::ULActionAE7, [1352](#)
 - gdcmm::network::ULActionAE8, [1354](#)
 - gdcmm::network::ULActionAR1, [1355](#)
 - gdcmm::network::ULActionAR10, [1356](#)
 - gdcmm::network::ULActionAR2, [1358](#)

- gdcmm::network::ULActionAR3, [1359](#)
- gdcmm::network::ULActionAR4, [1360](#)
- gdcmm::network::ULActionAR5, [1362](#)
- gdcmm::network::ULActionAR6, [1363](#)
- gdcmm::network::ULActionAR7, [1364](#)
- gdcmm::network::ULActionAR8, [1366](#)
- gdcmm::network::ULActionAR9, [1367](#)
- gdcmm::network::ULActionDT1, [1368](#)
- gdcmm::network::ULActionDT2, [1370](#)
- PerformedImagingAgentAdministrationSRStorage
 - gdcmm::UIDs, [1310](#)
- PET20StepColorPaletteSOPInstance
 - gdcmm::UIDs, [1308](#)
- PETColorPaletteSOPInstance
 - gdcmm::UIDs, [1308](#)
- PETImageStorage
 - gdcmm::MediaStorage, [800](#)
- PF
 - gdcmm::Bitmap, [262](#)
 - gdcmm::ImageCodec, [667](#)
- PGXCodec
 - gdcmm::PGXCodec, [925](#)
- PHILIPS
 - gdcmm::Dicts, [436](#)
- Philips3D
 - gdcmm::MediaStorage, [800](#)
- PhilipsPrivateMRSyntheticImageStorage
 - gdcmm::MediaStorage, [800](#)
- PhotometricInterpretation
 - gdcmm::PhotometricInterpretation, [928](#)
- PI
 - gdcmm::Bitmap, [262](#)
 - gdcmm::ImageCodec, [668](#)
- PI_END
 - gdcmm::PhotometricInterpretation, [928](#)
- PIType
 - gdcmm::PhotometricInterpretation, [927](#)
- PixelData
 - gdcmm::Bitmap, [262](#)
 - gdcmm::PixmapReader, [950](#)
 - gdcmm::PixmapWriter, [958](#)
- PixelFormat
 - gdcmm::PixelFormat, [933](#)
- Pixmap
 - gdcmm::Pixmap, [943](#)
- PixmapReader
 - gdcmm::Bitmap, [261](#)
 - gdcmm::PixmapReader, [949](#)
- PixmapToPixmapFilter
 - gdcmm::PixmapToPixmapFilter, [952](#)
- PixmapWriter
 - gdcmm::PixmapWriter, [956](#)
- PlanarConfiguration
 - gdcmm::Bitmap, [262](#)
- gdcmm::ImageCodec, [668](#)
- vtkGDCMImageReader, [1459](#)
- vtkGDCMImageReader2, [1473](#)
- PlannedImagingAgentAdministrationSRStorage
 - gdcmm::UIDs, [1310](#)
- PMS
 - gdcmm::EquipmentManufacturer, [531](#)
- PN
 - gdcmm::VR, [1433](#)
- PNComp
 - gdcmm, [87](#)
- PNMCodec
 - gdcmm::PNMCodec, [961](#)
- pointer
 - gdcmm::CodeString, [314](#)
 - gdcmm::LO, [775](#)
 - gdcmm::String< TDelimiter, TMaxLength, TPad-Char >, [1191](#)
- POINTS
 - gdcmm::Surface, [1205](#)
- Position
 - gdcmm::MrProtocol::Slice, [1135](#)
- PositronEmissionTomographyImageStorage
 - gdcmm::UIDs, [1306](#)
- Preamble
 - gdcmm::Preamble, [964](#)
- PrepareWrite
 - gdcmm::PixmapWriter, [957](#)
 - gdcmm::SegmentWriter, [1090](#)
 - gdcmm::SurfaceWriter, [1225](#)
- PrepareWritePointMacro
 - gdcmm::SurfaceWriter, [1225](#)
- Prepend
 - gdcmm::Global, [618](#)
- PresentationContext
 - gdcmm::PresentationContext, [968](#)
- PresentationContextAC
 - gdcmm::network::PresentationContextAC, [971](#)
- PresentationContextArrayType
 - gdcmm::network::AAssociateRQPDU, [124](#)
 - gdcmm::PresentationContextGenerator, [974](#)
- PresentationContextGenerator
 - gdcmm::PresentationContextGenerator, [974](#)
- PresentationContextRQ
 - gdcmm::network::PresentationContextRQ, [976](#), [977](#)
- PresentationDataValue
 - gdcmm::network::PresentationDataValue, [980](#)
- PresentationLUTSOPClass
 - gdcmm::UIDs, [1303](#)
- Preserve
 - gdcmm::Cleaner, [302](#)
- PrettyPrintOff
 - gdcmm::JSON, [768](#)

- PrettyPrintOn
 - gdcmm::JSON, [769](#)
- PrimitiveData
 - gdcmm::MeshPrimitive, [817](#)
- PrimitivesData
 - gdcmm::MeshPrimitive, [814](#)
- PrimitiveType
 - gdcmm::MeshPrimitive, [817](#)
- Print
 - gdcmm::ApplicationEntity, [148](#)
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [165](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [180](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [192](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [200](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [218](#)
 - gdcmm::BaseQuery, [234](#)
 - gdcmm::Bitmap, [256](#)
 - gdcmm::BoxRegion, [269](#)
 - gdcmm::ByteValue, [281](#)
 - gdcmm::CSAHeader, [353](#)
 - gdcmm::Curve, [368](#)
 - gdcmm::DataSet, [396](#)
 - gdcmm::DictPrinter, [434](#)
 - gdcmm::DirectionCosines, [443](#)
 - gdcmm::Directory, [447](#)
 - gdcmm::DPath, [452](#)
 - gdcmm::Element< TVR, TVM >, [460](#)
 - gdcmm::Element< TVR, VM::VM1_2 >, [466](#)
 - gdcmm::Element< TVR, VM::VM1_n >, [471](#)
 - gdcmm::Element< TVR, VM::VM2_2n >, [478](#)
 - gdcmm::Element< TVR, VM::VM2_n >, [484](#)
 - gdcmm::Element< TVR, VM::VM3_3n >, [490](#)
 - gdcmm::Element< TVR, VM::VM3_4 >, [496](#)
 - gdcmm::Element< TVR, VM::VM3_n >, [502](#)
 - gdcmm::Element< VR::AS, VM::VM5 >, [506](#)
 - gdcmm::Element< VR::OB, VM::VM1 >, [511](#)
 - gdcmm::Element< VR::OW, VM::VM1 >, [516](#)
 - gdcmm::Event, [534](#)
 - gdcmm::Image, [635](#)
 - gdcmm::LookupTable, [783](#)
 - gdcmm::MEC_MR3, [795](#)
 - gdcmm::MrProtocol, [845](#)
 - gdcmm::network::AAAbortPDU, [114](#)
 - gdcmm::network::AAAssociateACPDU, [118](#)
 - gdcmm::network::AAAssociateRJPDU, [121](#)
 - gdcmm::network::AAAssociateRQPDU, [126](#)
 - gdcmm::network::AbstractSyntax, [130](#)
 - gdcmm::network::ApplicationContext, [146](#)
 - gdcmm::network::AReleaseRPPDU, [150](#)
 - gdcmm::network::AReleaseRQPDU, [153](#)
 - gdcmm::network::AsynchronousOperationsWindowSub, [158](#)
 - gdcmm::network::BasePDU, [230](#)
 - gdcmm::network::ImplementationClassUIDSub, [698](#)
 - gdcmm::network::ImplementationVersionNameSub, [700](#)
 - gdcmm::network::MaximumLengthSub, [792](#)
 - gdcmm::network::PDataTFPDU, [907](#)
 - gdcmm::network::PresentationContextAC, [971](#)
 - gdcmm::network::PresentationContextRQ, [978](#)
 - gdcmm::network::PresentationDataValue, [981](#)
 - gdcmm::network::RoleSelectionSub, [1048](#)
 - gdcmm::network::ServiceClassApplicationInformation, [1115](#)
 - gdcmm::network::SOPClassExtendedNegotiationSub, [1141](#)
 - gdcmm::network::TransferSyntaxSub, [1276](#)
 - gdcmm::network::UserInformation, [1410](#)
 - gdcmm::Object, [874](#)
 - gdcmm::Orientation, [888](#)
 - gdcmm::Overlay, [896](#)
 - gdcmm::PDBHeader, [913](#)
 - gdcmm::PersonName, [920](#)
 - gdcmm::PixelFormat, [936](#)
 - gdcmm::Pixmap, [944](#)
 - gdcmm::Preamble, [965](#)
 - gdcmm::PresentationContext, [969](#)
 - gdcmm::Printer, [985](#)
 - gdcmm::Region, [1035](#)
 - gdcmm::Scanner, [1057](#)
 - gdcmm::Scanner2, [1067](#)
 - gdcmm::SegmentedPaletteColorLookupTable, [1081](#)
 - gdcmm::SequenceOfFragments, [1097](#)
 - gdcmm::SequenceOfItems, [1106](#)
 - gdcmm::Sorter, [1146](#)
 - gdcmm::StrictScanner, [1177](#)
 - gdcmm::StrictScanner2, [1187](#)
 - gdcmm::TagPath, [1257](#)
 - gdcmm::Testing, [1263](#)
 - gdcmm::Version, [1420](#)
 - gdcmm::XMLPrinter, [1569](#)
 - PrintASCII
 - gdcmm::ByteValue, [281](#)

- PrintASCIIXML
 - gdcm::ByteValue, 281
- PrintAsContinuousString
 - gdcm::Tag, 1252
- PrintAsContinuousUpperCaseString
 - gdcm::Tag, 1252
- PrintAsPipeSeparatedString
 - gdcm::Tag, 1252
- PrintDataElement
 - gdcm::Printer, 985
 - gdcm::XMLPrinter, 1569
- PrintDataElement2
 - gdcm::DictPrinter, 434
- PrintDataSet
 - gdcm::Printer, 986
 - gdcm::XMLPrinter, 1569
- PrintDataSet2
 - gdcm::DictPrinter, 434
- Printer
 - gdcm::Printer, 985
- PrinterConfigurationRetrievalSOPClass
 - gdcm::UIDs, 1303
- PrinterConfigurationRetrievalSOPInstance
 - gdcm::UIDs, 1303
- PrinterSOPClass
 - gdcm::UIDs, 1303
- PrinterSOPInstance
 - gdcm::UIDs, 1303
- PrintGroupLength
 - gdcm::ByteValue, 282
- PrintHex
 - gdcm::ByteValue, 282
- PrintHexXML
 - gdcm::ByteValue, 282
- PrintJobSOPClass
 - gdcm::UIDs, 1303
- PrintPNXML
 - gdcm::ByteValue, 282
- PrintQueueManagementSOPClassRetired
 - gdcm::UIDs, 1303
- PrintQueueSOPInstanceRetired
 - gdcm::UIDs, 1303
- PrintSelf
 - vtkGDCMImageReader, 1449
 - vtkGDCMImageReader2, 1464
 - vtkGDCMImageWriter, 1477
 - vtkGDCMMedicalImageProperties, 1484
 - vtkGDCMPolyDataReader, 1487
 - vtkGDCMPolyDataWriter, 1492
 - vtkGDCMTesting, 1497
 - vtkGDCMThreadedImageReader, 1501
 - vtkGDCMThreadedImageReader2, 1505
 - vtkImageColorViewer, 1516
 - vtkImageMapToColors16, 1525
 - vtkImageMapToWindowLevelColors2, 1531
 - vtkImagePlanarComponentsToComponents, 1534
 - vtkImageRGBToYBR, 1536
 - vtkImageYBRToRGB, 1539
 - vtkLookupTable16, 1542
 - vtkRTStructSetProperties, 1548
- PrintSQ
 - gdcm::Printer, 986
 - gdcm::XMLPrinter, 1569
- PrintStyle
 - gdcm::Printer, 987
 - gdcm::XMLPrinter, 1570
- PrintStyles
 - gdcm::Printer, 985
 - gdcm::XMLPrinter, 1568
- PrintTable
 - gdcm::network::ULTransitionTable, 1392
 - gdcm::Scanner, 1057
 - gdcm::Scanner2, 1068
 - gdcm::StrictScanner, 1177
 - gdcm::StrictScanner2, 1187
- PrintXML
 - gdcm::PrivateDict, 989
- PrivateBegin
 - gdcm::Scanner2, 1068
 - gdcm::StrictScanner2, 1187
- PrivateConstIterator
 - gdcm::Scanner2, 1062
 - gdcm::StrictScanner2, 1182
- PrivateDict
 - gdcm::PrivateDict, 988
- PrivateEnd
 - gdcm::Scanner2, 1068
 - gdcm::StrictScanner2, 1188
- PrivateMappingType
 - gdcm::Scanner2, 1062
 - gdcm::StrictScanner2, 1182
- PrivateTag
 - gdcm::PrivateTag, 993
- PrivateTagToValue
 - gdcm::Scanner2, 1062
 - gdcm::StrictScanner2, 1182
- PrivateTagToValueValueType
 - gdcm::Scanner2, 1062
 - gdcm::StrictScanner2, 1182
- ProceduralEventLoggingSOPClass
 - gdcm::UIDs, 1302
- ProceduralEventLoggingSOPInstance
 - gdcm::UIDs, 1302
- ProcedureLogStorage
 - gdcm::UIDs, 1305
- Process
 - gdcm::Parser, 903

- ProcessDataSet
 - gdcm::FileExplicitFilter, [570](#)
- ProcessPrivateTag
 - gdcm::Scanner2, [1068](#)
 - gdcm::StrictScanner2, [1188](#)
- ProcessPublicTag
 - gdcm::Scanner, [1057](#)
 - gdcm::Scanner2, [1068](#)
 - gdcm::StrictScanner, [1177](#)
 - gdcm::StrictScanner2, [1188](#)
- ProcessRequest
 - vtkGDCMImageReader2, [1464](#)
- ProduceCharacterSetDataElement
 - gdcm::QueryFactory, [1008](#)
- ProduceQuery
 - gdcm::QueryFactory, [1009](#)
- ProductCharacteristicsQuerySOPClass
 - gdcm::UIDs, [1307](#)
- ProgressEvent
 - gdcm::ProgressEvent, [997](#)
- PropertyCategory
 - gdcm::Segment, [1076](#)
- PropertyType
 - gdcm::Segment, [1077](#)
- PropertyTypeModifiers
 - gdcm::Segment, [1077](#)
- ProtocolApprovalInformationModelFIND
 - gdcm::UIDs, [1310](#)
- ProtocolApprovalInformationModelGET
 - gdcm::UIDs, [1310](#)
- ProtocolApprovalInformationModelMOVE
 - gdcm::UIDs, [1310](#)
- ProtocolApprovalStorage
 - gdcm::UIDs, [1310](#)
- PseudoColorSoftcopyPresentationStateStorageSOP-
Class
 - gdcm::UIDs, [1305](#)
- PubChemCompoundCID
 - gdcm::UIDs, [1308](#)
- PublicConstIterator
 - gdcm::Scanner2, [1062](#)
 - gdcm::StrictScanner2, [1182](#)
- PublicMappingType
 - gdcm::Scanner2, [1062](#)
 - gdcm::StrictScanner2, [1182](#)
- PublicTagToValue
 - gdcm::Scanner2, [1062](#)
 - gdcm::StrictScanner2, [1182](#)
- PublicTagToValueValueType
 - gdcm::Scanner2, [1063](#)
 - gdcm::StrictScanner2, [1182](#)
- PullPrintRequestSOPClassRetired
 - gdcm::UIDs, [1304](#)
- PullStoredPrintManagementMetaSOPClassRetired
 - gdcm::UIDs, [1304](#)
- Push
 - gdcm::TagPath, [1257](#)
- PushBackFile
 - vtkGDCMMedicalImageProperties, [1484](#)
- PVRGCodec
 - gdcm::PVRGCodec, [1002](#)
- Python Directory Reference, [67](#)
- PythonFilter
 - gdcm::PythonFilter, [1004](#)
- Quality
 - gdcm::JPEGCodec, [760](#)
- QueryFactory
 - gdcm::BaseQuery, [235](#)
 - gdcm::BaseRootQuery, [240](#)
 - gdcm::FindPatientRootQuery, [605](#)
 - gdcm::FindStudyRootQuery, [609](#)
 - gdcm::ModalityPerformedProcedureStepCreate-
Query, [821](#)
 - gdcm::ModalityPerformedProcedureStepSet-
Query, [824](#)
 - gdcm::MovePatientRootQuery, [839](#)
 - gdcm::MoveStudyRootQuery, [843](#)
 - gdcm::WLMFindQuery, [1558](#)
- RadiomicsOntology
 - gdcm::UIDs, [1308](#)
- RadiopharmaceuticalRadiationDoseSRStorage
 - gdcm::UIDs, [1310](#)
- RAWCodec
 - gdcm::RAWCodec, [1022](#)
- RawDataStorage
 - gdcm::MediaStorage, [800](#)
 - gdcm::UIDs, [1305](#)
- Read
 - gdcm::BasicOffsetTable, [247](#)
 - gdcm::ByteValue, [282](#)
 - gdcm::CommandDataSet, [323](#)
 - gdcm::CP246ExplicitDataElement, [334](#)
 - gdcm::DataElement, [378](#)
 - gdcm::DataSet, [396](#)
 - gdcm::Element< TVR, TVM >, [460](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [466](#)
 - gdcm::Element< TVR, VM::VM1_n >, [471](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [478](#)
 - gdcm::Element< TVR, VM::VM2_n >, [484](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [490](#)
 - gdcm::Element< TVR, VM::VM3_4 >, [496](#)
 - gdcm::Element< TVR, VM::VM3_n >, [502](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [506](#)
 - gdcm::Element< VR::OB, VM::VM1 >, [511](#)
 - gdcm::Element< VR::OW, VM::VM1 >, [516](#)
 - gdcm::EncodingImplementation< VR::VRASCII
>, [525](#)

- gdcm::EncodingImplementation< VR::VRBINARY
 >, 527
- gdcm::ExplicitDataElement, 542
- gdcm::ExplicitImplicitDataElement, 546
- gdcm::File, 551
- gdcm::FileMetaInformation, 579
- gdcm::Fragment, 613
- gdcm::ImageReader, 683
- gdcm::ImageRegionReader, 689
- gdcm::ImplicitDataElement, 704
- gdcm::Item, 725
- gdcm::network::AAabortPDU, 114
- gdcm::network::AAssociateACPDU, 118
- gdcm::network::AAssociateRJPDU, 121
- gdcm::network::AAssociateRQPDU, 126
- gdcm::network::AbstractSyntax, 130
- gdcm::network::ApplicationContext, 146
- gdcm::network::AReleaseRPPDU, 151
- gdcm::network::AReleaseRQPDU, 153
- gdcm::network::AsynchronousOperationsWin-
 dowSub, 158
- gdcm::network::BasePDU, 230
- gdcm::network::ImplementationClassUIDSub,
 698
- gdcm::network::ImplementationVersionName-
 Sub, 700
- gdcm::network::MaximumLengthSub, 792
- gdcm::network::PDataTFPDU, 907
- gdcm::network::PresentationContextAC, 972
- gdcm::network::PresentationContextRQ, 978
- gdcm::network::PresentationDataValue, 981
- gdcm::network::RoleSelectionSub, 1048
- gdcm::network::ServiceClassApplicationInfor-
 mation, 1115
- gdcm::network::SOPClassExtendedNegociation-
 Sub, 1141
- gdcm::network::TransferSyntaxSub, 1276
- gdcm::network::UserInformation, 1410
- gdcm::PGXCodec, 926
- gdcm::PixmapReader, 949
- gdcm::PNMCodec, 962
- gdcm::Preamble, 965
- gdcm::Reader, 1028
- gdcm::SegmentReader, 1085
- gdcm::SequenceOfFragments, 1098
- gdcm::SequenceOfItems, 1106
- gdcm::StreamImageReader, 1161
- gdcm::SurfaceReader, 1220
- gdcm::TableReader, 1243
- gdcm::Tag, 1252
- gdcm::UNExplicitDataElement, 1398
- gdcm::UNExplicitImplicitDataElement, 1402
- gdcm::ValueIO< TDE, TSwap, TType >, 1417
- gdcm::VL, 1423
- gdcm::VR, 1436
- gdcm::VR16ExplicitDataElement, 1440
- gdcm::VRVLSIZE< 0 >, 1443
- gdcm::VRVLSIZE< 1 >, 1444
- Read16
 - gdcm::VL, 1423
- ReadACRNEMAIImage
 - gdcm::ImageReader, 684
 - gdcm::PixmapReader, 949
- ReadBacktrack
 - gdcm::Fragment, 613
- ReadCompat
 - gdcm::FileMetaInformation, 579
- ReadCompatInternal
 - gdcm::FileMetaInformation, 579
- ReadComputeLength
 - gdcm::EncodingImplementation< VR::VRASCII
 >, 525
 - gdcm::EncodingImplementation< VR::VRBINARY
 >, 527
- ReadDataSet
 - gdcm::Reader, 1029
- Reader
 - gdcm::Reader, 1027
- ReadFiles
 - vtkGDCMThreadedImageReader, 1502
- ReadFromCommaSeparatedString
 - gdcm::PrivateTag, 994
 - gdcm::Tag, 1252
- ReadFromContinuousString
 - gdcm::Tag, 1253
- ReadFromPipeSeparatedString
 - gdcm::Tag, 1253
- ReadImage
 - gdcm::ImageReader, 684
 - gdcm::PixmapReader, 950
- ReadImageInformation
 - gdcm::StreamImageReader, 1161
- ReadImageInternal
 - gdcm::PixmapReader, 950
- ReadInformation
 - gdcm::ImageRegionReader, 689
- ReadInto
 - gdcm::network::PDataTFPDU, 907
 - gdcm::network::PresentationDataValue, 981
- ReadIntoBuffer
 - gdcm::ImageRegionReader, 689
- README.txt, 1575
- ReadMetaInformation
 - gdcm::Reader, 1029
- ReadNested
 - gdcm::DataSet, 397
- ReadNoSwap

- gdcm::EncodingImplementation< VR::VRASCII
>, [525](#)
 - gdcm::EncodingImplementation< VR::VRBINARY
>, [527](#)
- ReadOrSkip
 - gdcm::DataElement, [378](#)
- ReadPointMacro
 - gdcm::SurfaceReader, [1221](#)
- ReadPreamble
 - gdcm::Reader, [1029](#)
- ReadPreValue
 - gdcm::CP246ExplicitDataElement, [334](#)
 - gdcm::DataElement, [378](#)
 - gdcm::ExplicitDataElement, [542](#)
 - gdcm::ExplicitImplicitDataElement, [546](#)
 - gdcm::Fragment, [613](#)
 - gdcm::ImplicitDataElement, [704](#)
 - gdcm::SequenceOfFragments, [1098](#)
 - gdcm::UNExplicitDataElement, [1398](#)
 - gdcm::UNExplicitImplicitDataElement, [1402](#)
 - gdcm::VR16ExplicitDataElement, [1440](#)
- ReadSegment
 - gdcm::SegmentReader, [1086](#)
- ReadSegments
 - gdcm::SegmentReader, [1086](#)
- ReadSelectedPrivateTags
 - gdcm::DataSet, [397](#)
 - gdcm::Reader, [1029](#)
- ReadSelectedPrivateTagsWithLength
 - gdcm::DataSet, [397](#)
- ReadSelectedTags
 - gdcm::DataSet, [397](#)
 - gdcm::Reader, [1029](#)
- ReadSelectedTagsWithLength
 - gdcm::DataSet, [397](#)
- ReadSurface
 - gdcm::SurfaceReader, [1221](#)
- ReadSurfaces
 - gdcm::SurfaceReader, [1221](#)
- Readuint16
 - gdcm::DictConverter, [426](#)
- ReadUpToTag
 - gdcm::DataSet, [398](#)
 - gdcm::Reader, [1030](#)
- ReadUpToTagWithLength
 - gdcm::DataSet, [398](#)
- ReadValue
 - gdcm::CP246ExplicitDataElement, [334](#)
 - gdcm::DataElement, [379](#)
 - gdcm::ExplicitDataElement, [542](#)
 - gdcm::ExplicitImplicitDataElement, [546](#)
 - gdcm::Fragment, [614](#)
 - gdcm::ImplicitDataElement, [704](#)
 - gdcm::SequenceOfFragments, [1098](#)
 - gdcm::UNExplicitDataElement, [1398](#)
 - gdcm::UNExplicitImplicitDataElement, [1402](#)
 - gdcm::VR16ExplicitDataElement, [1440](#)
- ReadValueWithLength
 - gdcm::DataElement, [379](#)
 - gdcm::ImplicitDataElement, [704](#)
- ReadVM
 - gdcm::DictConverter, [426](#)
- ReadVR
 - gdcm::DictConverter, [427](#)
- ReadWithLength
 - gdcm::CP246ExplicitDataElement, [334](#)
 - gdcm::DataElement, [379](#)
 - gdcm::DataSet, [398](#)
 - gdcm::ExplicitDataElement, [543](#)
 - gdcm::ExplicitImplicitDataElement, [546](#)
 - gdcm::ImplicitDataElement, [704](#)
 - gdcm::UNExplicitDataElement, [1398](#)
 - gdcm::VR16ExplicitDataElement, [1440](#)
- RealWorldValueIntercept
 - gdcm::RealWorldValueMappingContent, [1032](#)
- RealWorldValueMappingStorage
 - gdcm::UIDs, [1305](#)
- RealWorldValueSlope
 - gdcm::RealWorldValueMappingContent, [1033](#)
- RecommendedDisplayCIELabToRGB
 - gdcm::SurfaceHelper, [1215](#)
- RecurseDataSet
 - gdcm::Anonymizer, [141](#)
- RED
 - gdcm::LookupTable, [779](#)
- red
 - gdcm::terminal, [110](#)
- reference
 - gdcm::CodeString, [314](#)
 - gdcm::LO, [775](#)
 - gdcm::String< TDelimiter, TMaxLength, TPad-
Char >, [1191](#)
- ReferencedColorPrintManagementMetaSOPClass-
Retired
 - gdcm::UIDs, [1303](#)
- ReferencedGrayscalePrintManagementMetaSOP-
ClassRetired
 - gdcm::UIDs, [1303](#)
- ReferencedImageBoxSOPClassRetired
 - gdcm::UIDs, [1303](#)
- ReferenceFrameOfReferenceUID
 - vtkRTStructSetProperties, [1552](#)
- ReferenceSeriesInstanceUID
 - vtkRTStructSetProperties, [1552](#)
- Region
 - gdcm::Region, [1034](#)
- Register
 - gdcm::Object, [874](#)

- Remove
 - gdcmm::Anonymizer, [141](#)
 - gdcmm::Cleaner, [302](#)
 - gdcmm::DataSet, [398](#)
 - gdcmm::FileAnonymizer, [555](#)
 - gdcmm::Preamble, [966](#)
- RemoveAllGroupLength
 - gdcmm::Cleaner, [302](#)
- RemoveAllIllegal
 - gdcmm::Cleaner, [303](#)
- RemoveAllMissingPrivateCreator
 - gdcmm::Cleaner, [303](#)
- RemoveAllObservers
 - gdcmm::Subject, [1201](#)
- RemoveDictEntry
 - gdcmm::PrivateDict, [989](#)
- RemoveFile
 - gdcmm::System, [1235](#)
- RemoveGroupLength
 - gdcmm::Anonymizer, [141](#)
- RemoveItemByIndex
 - gdcmm::SequenceOfItems, [1107](#)
- RemoveMissingPrivateCreator
 - gdcmm::Cleaner, [303](#)
- RemoveObserver
 - gdcmm::Subject, [1201](#)
- RemoveOverlay
 - gdcmm::Pixmap, [944](#)
- RemovePrivateTags
 - gdcmm::Anonymizer, [141](#)
- RemoveRetired
 - gdcmm::Anonymizer, [142](#)
- Render
 - vtkImageColorViewer, [1516](#)
- Renderer
 - vtkImageColorViewer, [1522](#)
- RenderWindow
 - vtkImageColorViewer, [1522](#)
- Replace
 - gdcmm::Anonymizer, [142](#)
 - gdcmm::CommandDataSet, [323](#)
 - gdcmm::DataSet, [398](#)
 - gdcmm::FileAnonymizer, [556](#)
 - gdcmm::FileMetaInformation, [579](#)
- ReplaceCodeMeaning
 - gdcmm::Cleaner, [303](#)
- ReplaceEmpty
 - gdcmm::DataSet, [399](#)
- RequestData
 - vtkGDCMImageReader2, [1464](#)
 - vtkGDCMPolyDataReader, [1488](#)
 - vtkImageMapToColors16, [1525](#)
 - vtkImageMapToWindowLevelColors2, [1531](#)
- vtkImagePlanarComponentsToComponents, [1534](#)
- RequestData_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [1488](#)
- RequestData_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [1488](#)
- RequestDataCompat
 - vtkGDCMImageReader, [1450](#)
 - vtkGDCMImageReader2, [1464](#)
 - vtkGDCMThreadedImageReader, [1502](#)
- RequestInformation
 - vtkGDCMImageReader2, [1465](#)
 - vtkGDCMPolyDataReader, [1488](#)
 - vtkGDCMThreadedImageReader2, [1505](#)
 - vtkImageMapToColors16, [1525](#)
 - vtkImageMapToWindowLevelColors2, [1531](#)
- RequestInformation_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [1488](#)
- RequestInformation_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [1488](#)
- RequestInformationCompat
 - vtkGDCMImageReader, [1450](#)
 - vtkGDCMImageReader2, [1465](#)
- RequestPaddedCompositePixelCode
 - gdcmm::ImageCodec, [668](#)
- RequestPlanarConfiguration
 - gdcmm::ImageCodec, [668](#)
- Rescale
 - gdcmm::Rescaler, [1038](#)
- RescaleFunctionIntoBestFit
 - gdcmm::Rescaler, [1038](#)
- Rescaler
 - gdcmm::Rescaler, [1037](#)
- ReserveDataElement
 - gdcmm::FileStreamer, [597](#)
- ReserveGroupDataElement
 - gdcmm::FileStreamer, [597](#)
- reset
 - gdcmm::terminal, [110](#)
- ResetHandledDataSet
 - gdcmm::network::ULConnectionCallback, [1379](#)
- RespiratoryWaveformStorage
 - gdcmm::UIDs, [1309](#)
- RetrieveSOPInstanceUIDFromIndex
 - gdcmm::DirectoryHelper, [450](#)
- RetrieveSOPInstanceUIDFromZPosition
 - gdcmm::DirectoryHelper, [450](#)
- reverse
 - gdcmm::terminal, [110](#)
- reverse_iterator
 - gdcmm::CodeString, [314](#)
 - gdcmm::LO, [775](#)

- gdcmm::String< TDelimiter, TMaxLength, TPad-Char >, 1191
- RFC2557MIMEencapsulation
 - gdcmm::UIDs, 1302
- RGB
 - gdcmm::PhotometricInterpretation, 928
- RGB2YBR
 - gdcmm::ImageChangePhotometricInterpretation, 643
- RGBPixelsToRGBPlanes
 - gdcmm::ImageChangePlanarConfiguration, 648
- RGBPlanesToRGBPixels
 - gdcmm::ImageChangePlanarConfiguration, 648
- RGBToRecommendedDisplayCIELab
 - gdcmm::SurfaceHelper, 1216
- RGBToRecommendedDisplayGrayscale
 - gdcmm::SurfaceHelper, 1216
- RLE_COMPRESSION
 - vtkGDCMImageWriter, 1476
- RLECodec
 - gdcmm::RLECodec, 1043
- RLELossless
 - gdcmm::TransferSyntax, 1272
 - gdcmm::UIDs, 1302
- ROI
 - gdcmm::Overlay, 892
- RoleSelectionSub
 - gdcmm::network::RoleSelectionSub, 1047
- Round
 - gdcmm, 101
- roundat
 - gdcmm, 101
- RTBeamsDeliveryInstructionStorage
 - gdcmm::UIDs, 1311
- RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
 - gdcmm::UIDs, 1306
- RTBeamsTreatmentRecordStorage
 - gdcmm::UIDs, 1306
- RTBrachyApplicationSetupDeliveryInstructionStorage
 - gdcmm::UIDs, 1311
- RTBrachyTreatmentRecordStorage
 - gdcmm::UIDs, 1306
- RTConventionalMachineVerification
 - gdcmm::UIDs, 1311
- RTConventionalMachineVerificationSupplement74FrozenDraft
 - gdcmm::UIDs, 1306
- RTDoseStorage
 - gdcmm::MediaStorage, 800
 - gdcmm::UIDs, 1306
- RTImageStorage
 - gdcmm::MediaStorage, 800
- gdcmm::UIDs, 1306
- RTIonBeamsTreatmentRecordStorage
 - gdcmm::MediaStorage, 801
 - gdcmm::UIDs, 1306
- RTIonMachineVerification
 - gdcmm::UIDs, 1311
- RTIonMachineVerificationSupplement74FrozenDraft
 - gdcmm::UIDs, 1306
- RTIonPlanStorage
 - gdcmm::MediaStorage, 800
 - gdcmm::UIDs, 1306
- RTPhysicianIntentStorage
 - gdcmm::UIDs, 1310
- RTPlanStorage
 - gdcmm::MediaStorage, 800
 - gdcmm::UIDs, 1306
- RTSegmentAnnotationStorage
 - gdcmm::UIDs, 1310
- RTStructSetProperties
 - vtkGDCMPolyDataReader, 1490
 - vtkGDCMPolyDataWriter, 1494
- RTStructureSetStorage
 - gdcmm::MediaStorage, 800
 - gdcmm::UIDs, 1306
- RTTreatmentSummaryRecordStorage
 - gdcmm::MediaStorage, 801
 - gdcmm::UIDs, 1306
- Rule
 - gdcmm::SerieHelper, 1110
- RunEventLoop
 - gdcmm::network::ULConnectionManager, 1386
- RunMoveEventLoop
 - gdcmm::network::ULConnectionManager, 1386
- SAGITTAL
 - gdcmm::Orientation, 887
- SAMSUNG
 - gdcmm::EquipmentManufacturer, 531
- ScalarType
 - gdcmm::PixelFormat, 932
- Scale
 - vtkGDCMImageReader, 1459
 - vtkGDCMImageReader2, 1473
- Scan
 - gdcmm::Scanner, 1057
 - gdcmm::Scanner2, 1068
 - gdcmm::StrictScanner, 1177
 - gdcmm::StrictScanner2, 1188
- Scanner
 - gdcmm::Scanner, 1053
- Scanner2
 - gdcmm::Scanner2, 1063
- Scrub
 - gdcmm::Cleaner, 303, 304

- SecondaryCaptureImageStorage
 - gdcmm::MediaStorage, 799
 - gdcmm::UIDs, 1304
- Segment
 - gdcmm::Segment, 1072
- SegmentAlgorithmName
 - gdcmm::Segment, 1077
- SegmentAlgorithmType
 - gdcmm::Segment, 1077
- Segmentation
 - gdcmm::MediaStorage, 802
- SegmentationStorage
 - gdcmm::MediaStorage, 800
 - gdcmm::UIDs, 1305
- SegmentDescription
 - gdcmm::Segment, 1077
- SegmentedPaletteColorLookupTable
 - gdcmm::SegmentedPaletteColorLookupTable, 1081
- SegmentedVolumeRenderingVolumetricPresentation-
StateStorage
 - gdcmm::UIDs, 1309
- SegmentLabel
 - gdcmm::Segment, 1077
- SegmentMap
 - gdcmm::SegmentReader, 1085
- SegmentNumber
 - gdcmm::Segment, 1077
- SegmentReader
 - gdcmm::SegmentReader, 1085
- Segments
 - gdcmm::SegmentReader, 1086
 - gdcmm::SegmentWriter, 1091
- SegmentVector
 - gdcmm::SegmentReader, 1085
 - gdcmm::SegmentWriter, 1090
- SegmentWriter
 - gdcmm::SegmentWriter, 1090
- Selection
 - gdcmm::Sorter, 1147
- SelectionMap
 - gdcmm::Sorter, 1145
- Self
 - gdcmm::AnonymizeEvent, 133
 - gdcmm::DataEvent, 386
 - gdcmm::DataSetEvent, 402
 - gdcmm::FileNameEvent, 586
 - gdcmm::MemberCommand< T >, 809
 - gdcmm::ProgressEvent, 997
 - gdcmm::SimpleMemberCommand< T >, 1129
- SEMIAUTOMATIC
 - gdcmm::Segment, 1072
- SendEcho
 - gdcmm::network::ULConnectionManager, 1386
- gdcmm::ServiceClassUser, 1120
- SendFind
 - gdcmm::network::ULConnectionManager, 1386, 1387
 - gdcmm::ServiceClassUser, 1120
- SendMove
 - gdcmm::network::ULConnectionManager, 1387
 - gdcmm::ServiceClassUser, 1120, 1121
- SendNAction
 - gdcmm::network::ULConnectionManager, 1387
- SendNCreate
 - gdcmm::network::ULConnectionManager, 1387
- SendNDelete
 - gdcmm::network::ULConnectionManager, 1388
- SendNEventReport
 - gdcmm::network::ULConnectionManager, 1388
- SendNGet
 - gdcmm::network::ULConnectionManager, 1388
- SendNSet
 - gdcmm::network::ULConnectionManager, 1388
- SendStore
 - gdcmm::network::ULConnectionManager, 1389
 - gdcmm::ServiceClassUser, 1121
- Separator
 - gdcmm::ApplicationEntity, 149
 - gdcmm::PersonName, 922
- SequenceLengthField
 - gdcmm::SequenceOfItems, 1108
- SequenceOfFragments
 - gdcmm::SequenceOfFragments, 1095
- SequenceOfItems
 - gdcmm::SequenceOfItems, 1103
- SerieHelper
 - gdcmm::SerieHelper, 1111
- SerieRestrictions
 - gdcmm::SerieHelper, 1110
- Series
 - gdcmm::Series, 1114
- SeriesInstanceUID
 - vtkRTStructSetProperties, 1552
- ServiceClassApplicationInformation
 - gdcmm::network::ServiceClassApplicationInformation, 1114
- ServiceClassUser
 - gdcmm::ServiceClassUser, 1119
- Set
 - gdcmm::Attribute< Group, Element, TVR, TVM >, 165
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 174
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185

- gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [192](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, [200](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, [218](#)
- gdcm::Element< TVR, TVM >, [460](#)
- gdcm::Element< TVR, VM::VM1_2 >, [466](#)
- gdcm::Element< TVR, VM::VM1_n >, [471](#)
- gdcm::Element< TVR, VM::VM2_2n >, [479](#)
- gdcm::Element< TVR, VM::VM2_n >, [484](#)
- gdcm::Element< TVR, VM::VM3_3n >, [491](#)
- gdcm::Element< TVR, VM::VM3_4 >, [496](#)
- gdcm::Element< TVR, VM::VM3_n >, [502](#)
- gdcm::Element< VR::AS, VM::VM5 >, [506](#)
- gdcm::Element< VR::OB, VM::VM1 >, [511](#)
- gdcm::Element< VR::OW, VM::VM1 >, [516](#)
- SetAbstractSyntax
 - gdcm::network::PresentationContextRQ, [978](#)
 - gdcm::PresentationContext, [969](#)
- SetAETitle
 - gdcm::ServiceClassUser, [1122](#)
- SetAlgorithmFamily
 - gdcm::Surface, [1210](#)
- SetAlgorithmName
 - gdcm::Surface, [1210](#)
- SetAlgorithmVersion
 - gdcm::Surface, [1210](#)
- SetAnatomicRegion
 - gdcm::Segment, [1075](#)
- SetAnatomicRegionModifiers
 - gdcm::Segment, [1075](#)
- SetAppendDerivationHistory
 - gdcm::FileDerivation, [567](#)
- SetArray
 - gdcm::Element< TVR, VM::VM1_n >, [472](#)
- setAttribute
 - gdcm::terminal, [111](#)
- SetAxisOfRotation
 - gdcm::Surface, [1210](#)
- setbgcolor
 - gdcm::terminal, [111](#)
- SetBitPosition
 - gdcm::Overlay, [896](#)
- SetBitsAllocated
 - gdcm::Overlay, [896](#)
 - gdcm::PixelFormat, [937](#)
- SetBitSample
 - gdcm::JPEGCodec, [758](#)
- SetBitsStored
 - gdcm::PixelFormat, [937](#)
- SetBlob
 - gdcm::ApplicationEntity, [148](#)
 - gdcm::network::PresentationDataValue, [981](#)
 - gdcm::PersonName, [921](#)
- SetBlueLUT
 - gdcm::LookupTable, [783](#)
- SetBufferLength
 - gdcm::JPEGLSCodec, [766](#)
 - gdcm::PNMCodec, [962](#)
 - gdcm::RLECodec, [1046](#)
- SetByteSwapTag
 - gdcm::ByteSwapFilter, [275](#)
- SetByteValue
 - gdcm::Attribute< Group, Element, TVR, TVM >, [165](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, [180](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [192](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, [200](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, [218](#)
 - gdcm::CSAElement, [346](#)
 - gdcm::DataElement, [379](#)
- SetByteValueNoSwap
 - gdcm::Attribute< Group, Element, TVR, TVM >, [165](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, [180](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, [185](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [193](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, [200](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, [205](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, [212](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, [218](#)
- SetCallbackFunction

- gdcmmembercommand< T >, 810, 811
- gdcmsimplemembercommand< T >, 1130
- SetCalledAETitle
 - gdcmmembercommand::AAssociateACPDU, 119
 - gdcmmembercommand::AAssociateRQPDU, 126
 - gdcmmembercommand::ServiceClassUser, 1122
- SetCallingAETitle
 - gdcmmembercommand::AAssociateACPDU, 119
 - gdcmmembercommand::AAssociateRQPDU, 126
- SetCenterOfRotation
 - gdcmmembercommand::Surface, 1210
- SetChangePrivateTags
 - gdcmmembercommand::FileExplicitFilter, 570
- SetCheckFileMetaInformation
 - gdcmmembercommand::Writer, 1562
- SetCipherType
 - gdcmmembercommand::CAPICryptographicMessageSyntax, 288
 - gdcmmembercommand::CryptographicMessageSyntax, 340
 - gdcmmembercommand::OpenSSLCryptographicMessageSyntax, 880
 - gdcmmembercommand::OpenSSL7CryptographicMessageSyntax, 885
- SetColor
 - gdcmmembercommand::Printer, 986
- SetColorLevel
 - vtkImageColorViewer, 1516
- SetColorWindow
 - vtkImageColorViewer, 1516
- SetColumns
 - gdcmmembercommand::Bitmap, 256
 - gdcmmembercommand::Overlay, 896
- SetCommand
 - gdcmmembercommand::PresentationDataValue, 981
- SetComponents
 - gdcmmembercommand::PersonName, 921
- SetCompressIconImage
 - gdcmmembercommand::ImageChangeTransferSyntax, 654
- SetComputeZSpacing
 - gdcmmembercommand::IPPSorter, 717
- SetCoordinateStartValue
 - gdcmmembercommand::Curve, 368
- SetCoordinateStepValue
 - gdcmmembercommand::Curve, 368
- SetCryptographicMessageSyntax
 - gdcmmembercommand::Anonymizer, 143
- SetCurve
 - gdcmmembercommand::Curve, 368
 - vtkGDCMImageReader, 1450
 - vtkGDCMImageReader2, 1465
- SetCurveDataDescriptor
 - gdcmmembercommand::Curve, 368
- SetCurveDescription
 - gdcmmembercommand::Curve, 368
- SetData
 - gdcmmembercommand::DataEvent, 388
- SetDataElement
 - gdcmmembercommand::Bitmap, 256
- SetDataSet
 - gdcmmembercommand::File, 551
 - gdcmmembercommand::PresentationDataValue, 981
- SetDataSetTransferSyntax
 - gdcmmembercommand::FileMetaInformation, 579
- SetDataValueRepresentation
 - gdcmmembercommand::Curve, 368
- SetDebug
 - gdcmmembercommand::Trace, 1267
- SetDebugStream
 - gdcmmembercommand::Trace, 1267
- SetDefaultTransferSyntax
 - gdcmmembercommand::PresentationContextGenerator, 975
- SetDerivationCodeSequenceCodeValue
 - gdcmmembercommand::FileDerivation, 567
- SetDerivationDescription
 - gdcmmembercommand::FileDerivation, 567
- SetDescription
 - gdcmmembercommand::CSAHeaderDictEntry, 359
 - gdcmmembercommand::ModuleEntry, 832
 - gdcmmembercommand::Overlay, 896
- SetDescriptor
 - gdcmmembercommand::DICOMDIRGenerator, 418
- SetDictName
 - gdcmmembercommand::DictConverter, 427
- SetDicts
 - gdcmmembercommand::PythonFilter, 1004
 - gdcmmembercommand::StringFilter, 1195
- SetDimension
 - gdcmmembercommand::Bitmap, 257
- SetDimensions
 - gdcmmembercommand::Bitmap, 257
 - gdcmmembercommand::Curve, 368
 - gdcmmembercommand::ImageCodec, 664
- SetDimensionsValue
 - gdcmmembercommand::ImageHelper, 678
- SetDirectionCosines
 - gdcmmembercommand::Image, 635
 - vtkGDCMImageWriter, 1477
- SetDirectionCosinesFromImageOrientationPatient
 - vtkGDCMImageWriter, 1477
- SetDirectionCosinesTolerance
 - gdcmmembercommand::IPPSorter, 718
- SetDirectionCosinesValue
 - gdcmmembercommand::ImageHelper, 678
- SetDirectory
 - gdcmmembercommand::network::ULWritingCallback, 1394
 - gdcmmembercommand::SerieHelper, 1112
- SetDisplayId
 - vtkImageColorViewer, 1516
- SetDomain

- gdcmm::BoxRegion, 269
- SetDropDuplicatePositions
 - gdcmm::IPPSorter, 718
- SetElement
 - gdcmm::Tag, 1253
- SetElementHandler
 - gdcmm::Parser, 903
- SetElementTag
 - gdcmm::Tag, 1253, 1254
- SetElementXX
 - gdcmm::DictEntry, 430
- SetError
 - gdcmm::Trace, 1267
- SetErrorStream
 - gdcmm::Trace, 1267
- SetEvent
 - gdcmm::network::ULEvent, 1391
- setfgcolor
 - gdcmm::terminal, 111
- SetFile
 - gdcmm::Anonymizer, 143
 - gdcmm::Cleaner, 304
 - gdcmm::DICOMDIRGenerator, 418
 - gdcmm::FileDecompressLookupTable, 564
 - gdcmm::FileDerivation, 567
 - gdcmm::FileExplicitFilter, 570
 - gdcmm::IconImageFilter, 624
 - gdcmm::Printer, 986
 - gdcmm::PythonFilter, 1004
 - gdcmm::Reader, 1030
 - gdcmm::SplitMosaicFilter, 1154
 - gdcmm::StreamImageWriter, 1165
 - gdcmm::StringFilter, 1195
 - gdcmm::Validate, 1413
 - gdcmm::Writer, 1562
 - gdcmm::XMLPrinter, 1569
- SetFileName
 - gdcmm::FileNameEvent, 587
 - gdcmm::Reader, 1030
 - gdcmm::StreamImageReader, 1161
 - gdcmm::StreamImageWriter, 1166
 - gdcmm::Writer, 1562
 - vtkGDCMThreadedImageReader2, 1505
- SetFilename
 - gdcmm::TableReader, 1243
- SetFileNames
 - vtkGDCMImageReader, 1450
 - vtkGDCMImageWriter, 1477
 - vtkGDCMThreadedImageReader2, 1505
- SetFileNames
 - gdcmm::DICOMDIRGenerator, 419
- SetFilePattern
 - vtkGDCMImageReader, 1450
 - vtkGDCMImageReader2, 1465
- SetFilePrefix
 - vtkGDCMImageReader, 1451
 - vtkGDCMImageReader2, 1465
- SetFiles
 - gdcmm::FileSet, 593
- SetFiniteVolume
 - gdcmm::Surface, 1210
- SetForce
 - gdcmm::ImageChangeTransferSyntax, 654
 - gdcmm::ImageFragmentSplitter, 673
- SetForcePixelSpacing
 - gdcmm::ImageHelper, 678
- SetForceRescaleInterceptSlope
 - gdcmm::ImageHelper, 678
- SetFragmentSizeMax
 - gdcmm::ImageFragmentSplitter, 673
- SetFrameOrigin
 - gdcmm::Overlay, 897
- SetFromDataElement
 - gdcmm::Attribute< Group, Element, TVR, TVM >, 166
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 174
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
 - gdcmm::Element< TVR, TVM >, 460
 - gdcmm::Element< TVR, VM::VM1_2 >, 466
 - gdcmm::Element< TVR, VM::VM1_n >, 472
 - gdcmm::Element< TVR, VM::VM2_2n >, 479
 - gdcmm::Element< TVR, VM::VM2_n >, 484
 - gdcmm::Element< TVR, VM::VM3_3n >, 491
 - gdcmm::Element< TVR, VM::VM3_4 >, 496
 - gdcmm::Element< TVR, VM::VM3_n >, 502
 - gdcmm::Element< VR::AS, VM::VM5 >, 506
 - gdcmm::Element< VR::OB, VM::VM1 >, 511
 - gdcmm::Element< VR::OW, VM::VM1 >, 516
- SetFromDataSet
 - gdcmm::Attribute< Group, Element, TVR, TVM >, 166
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 175
 - gdcmm::Attribute< Group, Element, TVR,

- VM::VM1_3 >, [180](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM1_8 >, [186](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM1_n >, [193](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM2_2n >, [200](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM2_n >, [206](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM3_3n >, [213](#)
- gdcm::Attribute< Group, Element, TVR,
 - VM::VM3_n >, [218](#)
- gdcm::MediaStorage, [804](#)
- SetFromFile
 - gdcm::MediaStorage, [804](#)
- SetFromHeader
 - gdcm::MediaStorage, [804](#)
- SetFromModality
 - gdcm::MediaStorage, [805](#)
- SetFromSourceImageSequence
 - gdcm::MediaStorage, [805](#)
- SetFromString
 - gdcm::DirectionCosines, [443](#)
- SetFromUID
 - gdcm::UIDs, [1328](#)
- SetGreenLUT
 - gdcm::LookupTable, [783](#)
- SetGroup
 - gdcm::Curve, [369](#)
 - gdcm::Overlay, [897](#)
 - gdcm::Tag, [1254](#)
- SetGroupXX
 - gdcm::DictEntry, [430](#)
- SetHeader
 - gdcm::File, [552](#)
- SetHighBit
 - gdcm::PixelFormat, [937](#)
- SetHostname
 - gdcm::ServiceClassUser, [1122](#)
- SetIconImage
 - gdcm::Pixmap, [945](#)
- SetIE
 - gdcm::IODEntry, [710](#)
- SetImage
 - gdcm::PixmapWriter, [957](#)
 - gdcm::SplitMosaicFilter, [1154](#)
- SetImplementationClassUID
 - gdcm::FileMetaInformation, [579](#)
- SetImplementationVersionName
 - gdcm::FileMetaInformation, [580](#)
- SetImplicitFlag
 - gdcm::network::ULConnectionCallback, [1379](#)
- SetInput
 - gdcm::BitmapToBitmapFilter, [264](#)
 - gdcm::ImageConverter, [669](#)
 - vtkImageColorViewer, [1516](#)
- SetInputConnection
 - vtkImageColorViewer, [1517](#)
- SetInputDirectory
 - gdcm::EmptyMaskGenerator, [521](#)
- SetInputFileName
 - gdcm::DictConverter, [427](#)
 - gdcm::FileAnonymizer, [556](#)
 - gdcm::FileChangeTransferSyntax, [560](#)
- SetIntercept
 - gdcm::Image, [635](#)
 - gdcm::Rescaler, [1039](#)
- SetKey
 - gdcm::CSAElement, [346](#)
- SetKeyword
 - gdcm::DictEntry, [430](#)
- SetLastElement
 - gdcm::ParseException, [900](#)
- SetLastFragment
 - gdcm::network::PresentationDataValue, [981](#)
- SetLength
 - gdcm::ByteValue, [282](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [466](#)
 - gdcm::Element< TVR, VM::VM1_n >, [472](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [479](#)
 - gdcm::Element< TVR, VM::VM2_n >, [484](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [491](#)
 - gdcm::Element< TVR, VM::VM3_4 >, [496](#)
 - gdcm::Element< TVR, VM::VM3_n >, [502](#)
 - gdcm::RLECodec, [1046](#)
 - gdcm::SequenceOfFragments, [1098](#)
 - gdcm::SequenceOfItems, [1107](#)
 - gdcm::Value, [1416](#)
- SetLengthOnly
 - gdcm::ByteValue, [283](#)
 - gdcm::Value, [1416](#)
- SetLengthToUndefined
 - gdcm::SequenceOfItems, [1107](#)
- SetLoadMode
 - gdcm::SerieHelper, [1113](#)
- SetLookupTable
 - vtkImageMapToColors16, [1525](#)
- SetLossless
 - gdcm::JPEGCodec, [758](#)
 - gdcm::JPEGLSCodec, [766](#)
- SetLossyError
 - gdcm::JPEGLSCodec, [766](#)
- SetLossyFlag
 - gdcm::Bitmap, [257](#)
 - gdcm::ImageCodec, [664](#)
 - gdcm::PVRGCodec, [1003](#)
- SetLUT

- gdcmm::Bitmap, 257
- gdcmm::ImageCodec, 664
- gdcmm::LookupTable, 783
- gdcmm::SegmentedPaletteColorLookupTable, 1081
- SetManifold
 - gdcmm::Surface, 1210
- SetMaximumLength
 - gdcmm::network::MaximumLengthSub, 792
- SetMaximumPointDistance
 - gdcmm::Surface, 1210
- SetMaxPDULength
 - gdcmm::network::ULConnectionInfo, 1381
- SetMaxPDUSize
 - gdcmm::network::ULConnection, 1376
- SetMCT
 - gdcmm::JPEG2000Codec, 744
- SetMeanPointDistance
 - gdcmm::Surface, 1211
- SetMedicalImageProperties
 - vtkGDCMImageReader, 1451
 - vtkGDCMImageReader2, 1465
 - vtkGDCMImageWriter, 1478
 - vtkGDCMPolyDataWriter, 1493
- SetMergeModeToAbstractSyntax
 - gdcmm::PresentationContextGenerator, 975
- SetMergeModeToTransferSyntax
 - gdcmm::PresentationContextGenerator, 975
- SetMeshPrimitive
 - gdcmm::Surface, 1211
- SetMessageHeader
 - gdcmm::network::PresentationDataValue, 982
- SetMinMaxForPixelType
 - gdcmm::Rescaler, 1039
- setmode
 - gdcmm::terminal, 111
- SetName
 - gdcmm::CSAElement, 346
 - gdcmm::CSAHeaderDictEntry, 359
 - gdcmm::DictEntry, 430
 - gdcmm::IODEntry, 710
 - gdcmm::Macro, 788
 - gdcmm::Module, 828
 - gdcmm::ModuleEntry, 832
 - gdcmm::network::AbstractSyntax, 130
 - gdcmm::network::ApplicationContext, 146
 - gdcmm::network::TransferSyntaxSub, 1276
 - gdcmm::PDBelement, 910
- SetNameFromUID
 - gdcmm::network::AbstractSyntax, 130
 - gdcmm::network::TransferSyntaxSub, 1276
- SetNeedByteSwap
 - gdcmm::Bitmap, 258
 - gdcmm::ImageCodec, 665
- SetNeedOverlayCleanup
 - gdcmm::ImageCodec, 665
- SetNestedDataSet
 - gdcmm::Item, 725
- SetNoOfItems
 - gdcmm::CSAElement, 346
- SetNoSwap
 - gdcmm::Element< TVR, TVM >, 461
 - gdcmm::Element< TVR, VM::VM1_2 >, 466
 - gdcmm::Element< TVR, VM::VM1_n >, 472
 - gdcmm::Element< TVR, VM::VM2_2n >, 479
 - gdcmm::Element< TVR, VM::VM2_n >, 484
 - gdcmm::Element< TVR, VM::VM3_3n >, 491
 - gdcmm::Element< TVR, VM::VM3_4 >, 496
 - gdcmm::Element< TVR, VM::VM3_n >, 502
 - gdcmm::Element< VR::AS, VM::VM5 >, 506
 - gdcmm::Element< VR::OB, VM::VM1 >, 511
 - gdcmm::Element< VR::OW, VM::VM1 >, 516
- SetNumberOfCurves
 - gdcmm::Pixmap, 945
- SetNumberOfDimensions
 - gdcmm::Bitmap, 258
 - gdcmm::ImageCodec, 665
- SetNumberOfFilenames
 - gdcmm::FilenameGenerator, 590
- SetNumberOfFrames
 - gdcmm::Overlay, 897
- SetNumberOfInputPorts
 - vtkGDCMPolyDataWriter, 1493
- SetNumberOfItems
 - gdcmm::SequenceOfItems, 1107
- SetNumberOfOverlays
 - gdcmm::Pixmap, 945
- SetNumberOfPoints
 - gdcmm::Curve, 369
- SetNumberOfResolutions
 - gdcmm::JPEG2000Codec, 744
- SetNumberOfSegments
 - gdcmm::SegmentWriter, 1091
- SetNumberOfSurfacePoints
 - gdcmm::Surface, 1211
- SetNumberOfSurfaces
 - gdcmm::SurfaceWriter, 1225
- SetNumberOfTableValues
 - vtkLookupTable16, 1542
- SetNumberOfThreadsForDecompression
 - gdcmm::JPEG2000Codec, 744
- SetNumberOfValues
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
- SetNumberOfVectors
 - gdcmm::Surface, 1211
- SetObliquityThresholdCosineValue
 - gdcmm::Orientation, 888

- SetOffScreenRendering
 - vtkImageColorViewer, [1517](#)
- SetOrigin
 - gdcm::Image, [635](#), [636](#)
 - gdcm::Overlay, [897](#)
- SetOriginValue
 - gdcm::ImageHelper, [679](#)
- SetOutputDimensions
 - gdcm::IconImageGenerator, [626](#)
- SetOutputDirectory
 - gdcm::EmptyMaskGenerator, [522](#)
- SetOutputFileName
 - gdcm::DictConverter, [427](#)
 - gdcm::FileAnonymizer, [556](#)
 - gdcm::FileChangeTransferSyntax, [560](#)
 - gdcm::FileStreamer, [597](#)
- SetOutputFormatToLuminance
 - vtkImageMapToColors16, [1525](#)
- SetOutputFormatToLuminanceAlpha
 - vtkImageMapToColors16, [1526](#)
- SetOutputFormatToRGB
 - vtkImageMapToColors16, [1526](#)
- SetOutputFormatToRGBA
 - vtkImageMapToColors16, [1526](#)
- SetOutputType
 - gdcm::DictConverter, [427](#)
- SetOutsideValuePixel
 - gdcm::IconImageGenerator, [626](#)
- SetOverlay
 - gdcm::Overlay, [897](#)
- SetOverlayVisibility
 - vtkImageColorViewer, [1517](#)
- SetOwner
 - gdcm::PrivateTag, [994](#)
- SetParentId
 - vtkImageColorViewer, [1517](#)
- SetPassword
 - gdcm::CAPICryptographicMessageSyntax, [288](#)
 - gdcm::CryptographicMessageSyntax, [340](#)
 - gdcm::OpenSSLCryptographicMessageSyntax, [880](#)
 - gdcm::OpenSSLP7CryptographicMessageSyntax, [885](#)
- SetPattern
 - gdcm::FilenameGenerator, [590](#)
- SetPDU
 - gdcm::network::ULEvent, [1391](#)
- SetPermissions
 - gdcm::System, [1235](#)
- SetPhotometricInterpretation
 - gdcm::Bitmap, [258](#)
 - gdcm::ImageChangePhotometricInterpretation, [644](#)
 - gdcm::ImageCodec, [665](#)
- SetPixelFormat
 - gdcm::Bitmap, [258](#)
 - gdcm::ImageCodec, [665](#)
 - gdcm::JPEGCodec, [758](#)
 - gdcm::Rescaler, [1039](#)
- SetPixelMinMax
 - gdcm::IconImageGenerator, [627](#)
- SetPixelRepresentation
 - gdcm::PixelFormat, [937](#)
- SetPixmap
 - gdcm::FileDecompressLookupTable, [564](#)
 - gdcm::IconImageGenerator, [627](#)
 - gdcm::PixmapWriter, [957](#)
- SetPlanarConfiguration
 - gdcm::Bitmap, [258](#)
 - gdcm::ImageChangePlanarConfiguration, [649](#)
 - gdcm::ImageCodec, [666](#)
- SetPMSRescaleInterceptSlope
 - gdcm::ImageHelper, [679](#)
- SetPointCoordinatesData
 - gdcm::Surface, [1211](#)
- SetPointPositionAccuracy
 - gdcm::Surface, [1211](#)
- SetPointsBoundingBoxCoordinates
 - gdcm::Surface, [1211](#)
- SetPort
 - gdcm::ServiceClassUser, [1122](#)
- SetPortSCP
 - gdcm::ServiceClassUser, [1122](#)
- SetPosition
 - vtkImageColorViewer, [1517](#)
- SetPreamble
 - gdcm::FileMetaInformation, [580](#)
- SetPrefix
 - gdcm::FilenameGenerator, [591](#)
- SetPresentationContextID
 - gdcm::network::PresentationContextAC, [972](#)
 - gdcm::network::PresentationContextRQ, [978](#)
 - gdcm::network::PresentationDataValue, [982](#)
 - gdcm::PresentationContext, [970](#)
- SetPresentationContexts
 - gdcm::network::ULConnection, [1376](#)
 - gdcm::ServiceClassUser, [1123](#)
- SetPrettyPrint
 - gdcm::JSON, [769](#)
- SetPrimitiveData
 - gdcm::MeshPrimitive, [816](#)
- SetPrimitivesData
 - gdcm::MeshPrimitive, [817](#)
- SetPrimitiveType
 - gdcm::MeshPrimitive, [817](#)
- SetPrivateCreator
 - gdcm::Tag, [1254](#)
- SetProcessingAlgorithm

- gdcmm::Surface, [1211](#)
- SetProgress
 - gdcmm::ProgressEvent, [998](#)
- SetPropertyCategory
 - gdcmm::Segment, [1075](#)
- SetPropertyType
 - gdcmm::Segment, [1075](#)
- SetPropertyTypeModifiers
 - gdcmm::Segment, [1075](#)
- SetPurposeOfReferenceCodeSequenceCodeValue
 - gdcmm::FileDerivation, [568](#)
- SetQuality
 - gdcmm::JPEG2000Codec, [744](#)
 - gdcmm::JPEGCodec, [759](#)
- SetRate
 - gdcmm::JPEG2000Codec, [744](#)
- SetReason
 - gdcmm::network::AAAbortPDU, [115](#)
 - gdcmm::network::PresentationContextAC, [972](#)
- SetRecommendedDisplayCIELabValue
 - gdcmm::Surface, [1212](#)
- SetRecommendedDisplayGrayscaleValue
 - gdcmm::Surface, [1212](#)
- SetRecommendedPresentationOpacity
 - gdcmm::Surface, [1212](#)
- SetRecommendedPresentationType
 - gdcmm::Surface, [1212](#)
- SetRecomputeItemLength
 - gdcmm::FileExplicitFilter, [570](#)
- SetRecomputeSequenceLength
 - gdcmm::FileExplicitFilter, [571](#)
- SetRedLUT
 - gdcmm::LookupTable, [783](#)
- SetRef
 - gdcmm::IODEntry, [711](#)
- SetRegion
 - gdcmm::ImageRegionReader, [690](#)
- SetRenderer
 - vtkImageColorViewer, [1517](#)
- SetRenderWindow
 - vtkImageColorViewer, [1518](#)
- SetRescaleInterceptSlopeValue
 - gdcmm::ImageHelper, [679](#)
- SetRetired
 - gdcmm::DictEntry, [430](#)
- SetReversible
 - gdcmm::JPEG2000Codec, [745](#)
- SetRGB8
 - gdcmm::ImageApplyLookupTable, [640](#)
- SetRoot
 - gdcmm::UIDGenerator, [1284](#)
- SetRootDirectory
 - gdcmm::DICOMDIRGenerator, [419](#)
- SetRows
 - gdcmm::Bitmap, [259](#)
 - gdcmm::Overlay, [897](#)
- SetRTStructSetProperties
 - vtkGDCMPolyDataWriter, [1493](#)
- SetSamplesPerPixel
 - gdcmm::PixelFormat, [937](#)
- SetScalarType
 - gdcmm::PixelFormat, [937](#)
- SetSearchParameter
 - gdcmm::BaseQuery, [234](#)
- SetSecondaryCaptureImagePlaneModule
 - gdcmm::ImageHelper, [679](#)
- SetSegmentAlgorithmName
 - gdcmm::Segment, [1075](#)
- SetSegmentAlgorithmType
 - gdcmm::Segment, [1075](#), [1076](#)
- SetSegmentDescription
 - gdcmm::Segment, [1076](#)
- SetSegmentLabel
 - gdcmm::Segment, [1076](#)
- SetSegmentNumber
 - gdcmm::Segment, [1076](#)
- SetSegments
 - gdcmm::SegmentWriter, [1091](#)
- SetSize
 - vtkImageColorViewer, [1518](#)
- SetSlice
 - vtkImageColorViewer, [1518](#)
- SetSliceOrientation
 - vtkImageColorViewer, [1518](#)
- SetSliceOrientationToXY
 - vtkImageColorViewer, [1518](#)
- SetSliceOrientationToXZ
 - vtkImageColorViewer, [1519](#)
- SetSliceOrientationToYZ
 - vtkImageColorViewer, [1519](#)
- SetSlope
 - gdcmm::Image, [636](#)
 - gdcmm::Rescaler, [1039](#)
- SetSOPClassUIDMode
 - gdcmm::EmptyMaskGenerator, [522](#)
- SetSOPInstanceUID
 - gdcmm::BaseQuery, [235](#)
- SetSortFunction
 - gdcmm::Sorter, [1146](#)
- SetSource
 - gdcmm::network::AAAbortPDU, [115](#)
- SetSourceApplicationEntityTitle
 - gdcmm::FileMetaInformation, [580](#)
- SetSpacing
 - gdcmm::Image, [636](#)
- SetSpacingValue
 - gdcmm::ImageHelper, [679](#)
- SetState

- gdcmm::network::ULConnection, 1377
- SetStream
 - gdcmm::Reader, 1031
 - gdcmm::StreamImageReader, 1162
 - gdcmm::StreamImageWriter, 1166
 - gdcmm::Trace, 1267
 - gdcmm::Writer, 1563
- SetStreamToFile
 - gdcmm::Trace, 1268
- SetStyle
 - gdcmm::Printer, 986
 - gdcmm::XMLPrinter, 1570
- SetSurfaceComments
 - gdcmm::Surface, 1212
- SetSurfaceCount
 - gdcmm::Segment, 1076
- SetSurfaceNumber
 - gdcmm::Surface, 1212
- SetSurfaceProcessing
 - gdcmm::Surface, 1213
- SetSurfaceProcessingDescription
 - gdcmm::Surface, 1213
- SetSurfaceProcessingRatio
 - gdcmm::Surface, 1213
- SetSyngoDT
 - gdcmm::CSAElement, 346
- SetTag
 - gdcmm::AnonymizeEvent, 134
 - gdcmm::DataElement, 380
- SetTagsToRead
 - gdcmm::Sorter, 1146
- SetTargetPixelType
 - gdcmm::Rescaler, 1039
- SetTemplateFileName
 - gdcmm::FileStreamer, 597
- SetTileSize
 - gdcmm::JPEG2000Codec, 745
- SetTimeout
 - gdcmm::network::ARTIMTimer, 155
 - gdcmm::ServiceClassUser, 1123
- SetToUndefined
 - gdcmm::VL, 1423
- SetTransferSyntax
 - gdcmm::Bitmap, 259
 - gdcmm::FileChangeTransferSyntax, 561
 - gdcmm::ImageChangeTransferSyntax, 654
 - gdcmm::network::PresentationContextAC, 972
- SetTuple
 - gdcmm::network::RoleSelectionSub, 1048
 - gdcmm::network::ServiceClassApplicationInformation, 1115
 - gdcmm::network::SOPClassExtendedNegociationSub, 1141
- SetType
 - gdcmm::ModuleEntry, 832
 - gdcmm::Overlay, 898
- SetTypeOfData
 - gdcmm::Curve, 369
- SetupInteractor
 - vtkImageColorViewer, 1519
- SetUsage
 - gdcmm::IODEntry, 711
- SetUserCodec
 - gdcmm::ImageChangeTransferSyntax, 654
- SetUserData
 - gdcmm::Parser, 904
- SetUserInformation
 - gdcmm::network::AAAssociateRQPDU, 126
- SetUseSeriesDetails
 - gdcmm::SerieHelper, 1113
- SetUseTargetPixelType
 - gdcmm::Rescaler, 1040
- SetUseVRUN
 - gdcmm::FileExplicitFilter, 571
- SetValue
 - gdcmm::Attribute< Group, Element, TVR, TVM >, 166
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 175
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 193, 194
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
 - gdcmm::CSAElement, 346
 - gdcmm::DataElement, 380
 - gdcmm::Element< TVR, TVM >, 461
 - gdcmm::Element< TVR, VM::VM1_2 >, 467
 - gdcmm::Element< TVR, VM::VM1_n >, 473
 - gdcmm::Element< TVR, VM::VM2_2n >, 479
 - gdcmm::Element< TVR, VM::VM2_n >, 485
 - gdcmm::Element< TVR, VM::VM3_3n >, 491
 - gdcmm::Element< TVR, VM::VM3_4 >, 497
 - gdcmm::Element< TVR, VM::VM3_n >, 503
 - gdcmm::Element< VR::AS, VM::VM5 >, 506
 - gdcmm::Element< VR::OB, VM::VM1 >, 511
 - gdcmm::Element< VR::OW, VM::VM1 >, 516
 - gdcmm::PDBElement, 910
- SetValueFieldLength

- gdcmm::DataElement, [381](#)
- SetValues
 - gdcmm::Attribute< Group, Element, TVR, TVM >, [167](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [175](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [180](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [186](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [194](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [200](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, [206](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, [213](#)
 - gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, [218](#)
- SetVectorAccuracy
 - gdcmm::Surface, [1213](#)
- SetVectorCoordinateData
 - gdcmm::Surface, [1213](#)
- SetVectorDimensionality
 - gdcmm::Surface, [1213](#)
- SetVL
 - gdcmm::DataElement, [381](#)
- SetVLToUndefined
 - gdcmm::DataElement, [381](#)
- SetVM
 - gdcmm::CSAElement, [347](#)
 - gdcmm::CSAHeaderDictEntry, [359](#)
 - gdcmm::DictEntry, [431](#)
- SetVR
 - gdcmm::CSAElement, [347](#)
 - gdcmm::CSAHeaderDictEntry, [359](#)
 - gdcmm::DataElement, [381](#)
 - gdcmm::DictEntry, [431](#)
- SetWarning
 - gdcmm::Trace, [1268](#)
- SetWarningStream
 - gdcmm::Trace, [1268](#)
- SetWindowId
 - vtkImageColorViewer, [1519](#)
- SetWriteDataSetOnly
 - gdcmm::Writer, [1563](#)
- SetZSpacingTolerance
 - gdcmm::IPPSorter, [718](#)
- SH
 - gdcmm::VR, [1433](#)
- SHA1
 - gdcmm::SHA1, [1125](#)
- SHComp
 - gdcmm, [88](#)
- Shift
 - vtkGDCMImageReader, [1459](#)
 - vtkGDCMImageReader2, [1473](#)
- ShiftEnd
 - gdcmm::ByteBuffer, [271](#)
- ShowAbort
 - gdcmm::SimpleSubjectWatcher, [1132](#)
- ShowAnonymization
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- ShowData
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- ShowDataSet
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- ShowFileName
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- ShowIteration
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- ShowProgress
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- SIEMENS
 - gdcmm::Dicts, [436](#)
 - gdcmm::EquipmentManufacturer, [531](#)
- SimpleMemberCommand
 - gdcmm::SimpleMemberCommand< T >, [1129](#)
- SimpleSubjectWatcher
 - gdcmm::SimpleSubjectWatcher, [1132](#)
- SimplifiedAdultEchoSRStorage
 - gdcmm::UIDs, [1310](#)
- SINGLEBIT
 - gdcmm::PixelFormat, [933](#)
- SingleSerieUIDFileSetHT
 - gdcmm::SerieHelper, [1113](#)
- SingleSerieUIDFileSetmap
 - gdcmm::SerieHelper, [1110](#)
- Size
 - gdcmm::CodeString, [315](#)
 - gdcmm::DataSet, [399](#)
 - gdcmm::GroupDict, [620](#)
 - gdcmm::network::AAAbortPDU, [115](#)
 - gdcmm::network::AAAssociateACPDU, [119](#)
 - gdcmm::network::AAAssociateRJPDU, [121](#)
 - gdcmm::network::AAAssociateRQPDU, [127](#)
 - gdcmm::network::AbstractSyntax, [130](#)
 - gdcmm::network::ApplicationContext, [146](#)
 - gdcmm::network::AReleaseRPPDU, [151](#)
 - gdcmm::network::AReleaseRQPDU, [153](#)
 - gdcmm::network::AsynchronousOperationsWindowSub, [158](#)
 - gdcmm::network::BasePDU, [230](#)
 - gdcmm::network::ImplementationClassUIDSub, [698](#)
 - gdcmm::network::ImplementationVersionNameSub, [700](#)

- gdcmm::network::MaximumLengthSub, 792
- gdcmm::network::PDataTFPDU, 907
- gdcmm::network::PresentationContextAC, 972
- gdcmm::network::PresentationContextRQ, 978
- gdcmm::network::PresentationDataValue, 982
- gdcmm::network::RoleSelectionSub, 1048
- gdcmm::network::ServiceClassApplicationInformation, 1115
- gdcmm::network::SOPClassExtendedNegociationSub, 1141
- gdcmm::network::TransferSyntaxSub, 1276
- gdcmm::network::UserInformation, 1410
- size_type
 - gdcmm::CodeString, 314
 - gdcmm::LO, 775
 - gdcmm::String< TDelimiter, TMaxLength, TPadChar >, 1191
- SizeType
 - gdcmm::DataSet, 391
 - gdcmm::FilenameGenerator, 589
 - gdcmm::IOD, 707
 - gdcmm::NestedModuleEntries, 857
 - gdcmm::network::AAssociateACPDU, 117
 - gdcmm::network::AAssociateRQPDU, 124
 - gdcmm::network::PDataTFPDU, 906
 - gdcmm::network::PresentationContextRQ, 976
 - gdcmm::PresentationContext, 968
 - gdcmm::PresentationContextGenerator, 974
 - gdcmm::SequenceOfFragments, 1094
 - gdcmm::SequenceOfItems, 1103
- SL
 - gdcmm::VR, 1433
- Slice
 - vtkImageColorViewer, 1522
- SLICE_ORIENTATION_XY
 - vtkImageColorViewer, 1513
- SLICE_ORIENTATION_XZ
 - vtkImageColorViewer, 1513
- SLICE_ORIENTATION_YZ
 - vtkImageColorViewer, 1513
- SliceOrientation
 - vtkImageColorViewer, 1522
- Slices
 - gdcmm::MrProtocol::SliceArray, 1136
- SmartPointer
 - gdcmm::Object, 874
 - gdcmm::SmartPointer< ObjectType >, 1138, 1139
- SOPClassExtendedNegociationSub
 - gdcmm::network::SOPClassExtendedNegociationSub, 1141
- SOPClassUIDMode
 - gdcmm::EmptyMaskGenerator, 521
- SOPInstanceUID
 - vtkRTStructSetProperties, 1552
- Sort
 - gdcmm::IPPSorter, 718
 - gdcmm::Sorter, 1147
- Sorter
 - gdcmm::Sorter, 1145
- SortFunc
 - gdcmm::Sorter, 1148
- SortFunction
 - gdcmm::Sorter, 1145
- Source Directory Reference, 67
- SpacialFiducialsStorage
 - gdcmm::MediaStorage, 800
- SpacialRegistrationStorage
 - gdcmm::MediaStorage, 800
- Spacing
 - gdcmm::Spacing, 1150
- SpacingType
 - gdcmm::Spacing, 1150
- SpatialFiducialsStorage
 - gdcmm::UIDs, 1305
- SpatialRegistrationStorage
 - gdcmm::UIDs, 1305
- SpectaclePrescriptionReportStorage
 - gdcmm::UIDs, 1309
- Spectroscopy
 - gdcmm::Spectroscopy, 1151
- Split
 - gdcmm::ImageFragmentSplitter, 673
 - gdcmm::SplitMosaicFilter, 1154
- SplitExtent
 - vtkGDCMThreadedImageReader2, 1506
- SplitMosaicFilter
 - gdcmm::SplitMosaicFilter, 1152
- SPM2AVG152PDFrameofReference
 - gdcmm::UIDs, 1302
- SPM2AVG152T1FrameofReference
 - gdcmm::UIDs, 1302
- SPM2AVG152T2FrameofReference
 - gdcmm::UIDs, 1302
- SPM2AVG305T1FrameofReference
 - gdcmm::UIDs, 1302
- SPM2BRAINMASKFrameofReference
 - gdcmm::UIDs, 1302
- SPM2CSFFrameofReference
 - gdcmm::UIDs, 1302
- SPM2EPIFrameofReference
 - gdcmm::UIDs, 1302
- SPM2FILT1FrameofReference
 - gdcmm::UIDs, 1302
- SPM2GRAYFrameofReference
 - gdcmm::UIDs, 1302
- SPM2PDFrameofReference
 - gdcmm::UIDs, 1302
- SPM2PETFrameofReference

- gdcmm::UIDs, [1302](#)
- SPM2SINGLESUBJT1FrameofReference
 - gdcmm::UIDs, [1302](#)
- SPM2SPECTFrameofReference
 - gdcmm::UIDs, [1302](#)
- SPM2T1FrameofReference
 - gdcmm::UIDs, [1302](#)
- SPM2T2FrameofReference
 - gdcmm::UIDs, [1302](#)
- SPM2TRANSMFrameofReference
 - gdcmm::UIDs, [1302](#)
- SPM2WHITEFrameofReference
 - gdcmm::UIDs, [1302](#)
- SpringColorPaletteSOPInstance
 - gdcmm::UIDs, [1308](#)
- SQ
 - gdcmm::VR, [1433](#)
- Squeeze
 - gdcmm::ApplicationEntity, [148](#)
- SS
 - gdcmm::VR, [1433](#)
- ST
 - gdcmm::VR, [1433](#)
- StableSort
 - gdcmm::Sorter, [1147](#)
- StandaloneCurveStorage
 - gdcmm::MediaStorage, [799](#)
- StandaloneCurveStorageRetired
 - gdcmm::UIDs, [1304](#)
- StandaloneModalityLUTStorage
 - gdcmm::MediaStorage, [799](#)
- StandaloneModalityLUTStorageRetired
 - gdcmm::UIDs, [1304](#)
- StandaloneOverlayStorage
 - gdcmm::MediaStorage, [799](#)
- StandaloneOverlayStorageRetired
 - gdcmm::UIDs, [1304](#)
- StandalonePETCurveStorageRetired
 - gdcmm::UIDs, [1306](#)
- StandaloneVOILUTStorage
 - gdcmm::MediaStorage, [799](#)
- StandaloneVOILUTStorageRetired
 - gdcmm::UIDs, [1304](#)
- Start
 - gdcmm::network::ARTIMTimer, [155](#)
- StartAssociation
 - gdcmm::ServiceClassUser, [1123](#)
- StartDataElement
 - gdcmm::FileStreamer, [598](#)
- StartElement
 - gdcmm::TableReader, [1243](#)
 - gdcmm::XMLDictReader, [1567](#)
 - gdcmm::XMLPrivateDictReader, [1573](#)
- StartElementHandler
 - gdcmm::Parser, [902](#)
- StartEncode
 - gdcmm::ImageCodec, [666](#)
 - gdcmm::JPEG2000Codec, [745](#)
 - gdcmm::JPEGCodec, [759](#)
 - gdcmm::JPEGLSCodec, [766](#)
 - gdcmm::RLECodec, [1046](#)
- StartFilter
 - gdcmm::SimpleSubjectWatcher, [1133](#)
- StartGroupDataElement
 - gdcmm::FileStreamer, [598](#)
- STATES
 - gdcmm::Surface, [1204](#)
- STATES_END
 - gdcmm::Surface, [1205](#)
- STComp
 - gdcmm, [88](#)
- StereometricRelationshipStorage
 - gdcmm::UIDs, [1305](#)
- Stop
 - gdcmm::network::ARTIMTimer, [155](#)
- StopAssociation
 - gdcmm::ServiceClassUser, [1123](#)
- StopDataElement
 - gdcmm::FileStreamer, [598](#)
- StopEncode
 - gdcmm::ImageCodec, [666](#)
 - gdcmm::JPEG2000Codec, [745](#)
 - gdcmm::JPEGCodec, [759](#)
 - gdcmm::JPEGLSCodec, [767](#)
 - gdcmm::RLECodec, [1046](#)
- StopGroupDataElement
 - gdcmm::FileStreamer, [598](#)
- StopProtocol
 - gdcmm::network::ULConnection, [1377](#)
- StorageCommitmentPullModelSOPClassRetired
 - gdcmm::UIDs, [1302](#)
- StorageCommitmentPullModelSOPInstanceRetired
 - gdcmm::UIDs, [1302](#)
- StorageCommitmentPushModelSOPClass
 - gdcmm::UIDs, [1302](#)
- StorageCommitmentPushModelSOPInstance
 - gdcmm::UIDs, [1302](#)
- StorageServiceClass
 - gdcmm::UIDs, [1303](#)
- StoredPrintStorageSOPClassRetired
 - gdcmm::UIDs, [1303](#)
- StrCaseCmp
 - gdcmm::System, [1235](#)
- Stream
 - gdcmm::Writer, [1564](#)
- StreamImageReader
 - gdcmm::Reader, [1031](#)
 - gdcmm::StreamImageReader, [1159](#)

- StreamImageWriter
 - gdcm::StreamImageWriter, [1164](#)
 - gdcm::Writer, [1564](#)
- StrictScanner
 - gdcm::StrictScanner, [1173](#)
- StrictScanner2
 - gdcm::StrictScanner2, [1183](#)
- String
 - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1192](#)
- StringFilter
 - gdcm::StringFilter, [1194](#)
- StrNCaseCmp
 - gdcm::System, [1235](#)
- StrSep
 - gdcm::System, [1236](#)
- StrTokR
 - gdcm::System, [1236](#)
- StructureSetDate
 - vtkRTStructSetProperties, [1552](#)
- StructureSetLabel
 - vtkRTStructSetProperties, [1552](#)
- StructureSetName
 - vtkRTStructSetProperties, [1553](#)
- StructureSetTime
 - vtkRTStructSetProperties, [1553](#)
- Study
 - gdcm::Study, [1197](#)
- StudyComponentManagementSOPClass
 - gdcm::MediaStorage, [800](#)
- StudyComponentManagementSOPClassRetired
 - gdcm::UIDs, [1303](#)
- StudyInstanceUID
 - vtkRTStructSetProperties, [1553](#)
- StudyRootQueryRetrieveInformationModelFIND
 - gdcm::UIDs, [1306](#)
- StudyRootQueryRetrieveInformationModelGET
 - gdcm::UIDs, [1306](#)
- StudyRootQueryRetrieveInformationModelMOVE
 - gdcm::UIDs, [1306](#)
- Subject
 - gdcm::Subject, [1199](#)
- SubjectiveRefractionMeasurementsStorage
 - gdcm::UIDs, [1309](#)
- SubstanceAdministrationLoggingSOPClass
 - gdcm::UIDs, [1302](#)
- SubstanceAdministrationLoggingSOPInstance
 - gdcm::UIDs, [1302](#)
- SubstanceApprovalQuerySOPClass
 - gdcm::UIDs, [1307](#)
- SummerColorPaletteSOPInstance
 - gdcm::UIDs, [1308](#)
- Superclass
 - gdcm::AnonymizeEvent, [133](#)
- gdcm::DataEvent, [386](#)
- gdcm::DataSetEvent, [402](#)
- gdcm::FileNameEvent, [586](#)
- gdcm::LO, [775](#)
- gdcm::ProgressEvent, [997](#)
- SURFACE
 - gdcm::Surface, [1205](#)
- Surface
 - gdcm::Surface, [1205](#)
- SurfaceCount
 - gdcm::Segment, [1077](#)
- SurfaceReader
 - gdcm::SurfaceReader, [1220](#)
- Surfaces
 - gdcm::Segment, [1077](#)
- SurfaceScanMeshStorage
 - gdcm::UIDs, [1309](#)
- SurfaceScanPointCloudStorage
 - gdcm::UIDs, [1309](#)
- SurfaceSegmentationStorage
 - gdcm::MediaStorage, [801](#)
 - gdcm::UIDs, [1308](#)
- SurfaceVector
 - gdcm::Segment, [1072](#)
- SurfaceWriter
 - gdcm::SurfaceWriter, [1224](#)
- SV
 - gdcm::VR, [1433](#)
- SV10
 - gdcm::CSAHeader, [350](#)
- Swap
 - gdcm::ByteSwap< T >, [272](#)
 - gdcm::SwapperDoOp, [1228](#)
 - gdcm::SwapperNoOp, [1229](#)
- SwapArray
 - gdcm::SwapperDoOp, [1228](#)
 - gdcm::SwapperNoOp, [1229](#)
- SwapCode
 - gdcm::SwapCode, [1227](#)
- SwapCodeType
 - gdcm::SwapCode, [1226](#)
- SwapFromSwapCodeIntoSystem
 - gdcm::ByteSwap< T >, [272](#)
- SwapRange
 - gdcm::ByteSwap< T >, [272](#)
- SwapRangeFromSwapCodeIntoSystem
 - gdcm::ByteSwap< T >, [272](#)
- SyngoDTField
 - gdcm::CSAElement, [348](#)
- SyntaxError
 - gdcm::Parser, [902](#)
- SystemIsBigEndian
 - gdcm::ByteSwap< T >, [273](#)
- SystemIsLittleEndian

- gdcmm::ByteSwap< T >, 273
- T1
 - gdcmm::Type, 1280
- T1C
 - gdcmm::Type, 1280
- T2
 - gdcmm::Type, 1280
- T2C
 - gdcmm::Type, 1280
- T3
 - gdcmm::Type, 1280
- Table
 - gdcmm::Table, 1238
- Table16
 - vtkLookupTable16, 1543
- TableEntry
 - gdcmm::TableEntry, 1240
- TableInternal
 - gdcmm::Table, 1239
- TableReader
 - gdcmm::TableReader, 1241
- TableRow
 - gdcmm::network::TableRow, 1244
- Tag
 - gdcmm::Tag, 1248
- tag
 - gdcmm::Tag, 1255
- TagField
 - gdcmm::DataElement, 383
- TagMismatchError
 - gdcmm::Parser, 902
- TagPath
 - gdcmm::TagPath, 1256
- tags
 - gdcmm::Tag, 1255
- TagsToRead
 - gdcmm::Sorter, 1148
- TagToValue
 - gdcmm::Scanner, 1052
 - gdcmm::StrictScanner, 1172
- TagToValueValueType
 - gdcmm::Scanner, 1052
 - gdcmm::StrictScanner, 1173
- TalairachBrainAtlasFrameofReference
 - gdcmm::UIDs, 1302
- TConstMemberFunctionPointer
 - gdcmm::MemberCommand< T >, 809
- TestAbortOff
 - gdcmm::SimpleSubjectWatcher, 1134
- TestAbortOn
 - gdcmm::SimpleSubjectWatcher, 1134
- Testing
 - gdcmm::Testing, 1259
- TestPBKDF2
 - gdcmm::ASN1, 157
- TestsList.txt, 1575
- TextSRStorageTrialRetired
 - gdcmm::UIDs, 1305
- ThreadedExecute
 - vtkImageRGBToYBR, 1536
 - vtkImageYBRToRGB, 1539
- ThreadedRequestData
 - vtkGDCMThreadedImageReader2, 1506
 - vtkImageMapToColors16, 1526
 - vtkImageMapToWindowLevelColors2, 1531
- TM
 - gdcmm::VR, 1433
- TMComp
 - gdcmm, 88
- TMemberFunctionPointer
 - gdcmm::MemberCommand< T >, 809
 - gdcmm::SimpleMemberCommand< T >, 1129
- Todo List, 3
- ToPyObject
 - gdcmm::PythonFilter, 1005
- TOSHIBA
 - gdcmm::EquipmentManufacturer, 531
- ToshibaPrivateDataStorage
 - gdcmm::MediaStorage, 800
- ToString
 - gdcmm::StringFilter, 1196
- ToStringPair
 - gdcmm::StringFilter, 1196, 1197
- ToUnixSlashes
 - gdcmm::Filename, 583
- ToWindowsSlashes
 - gdcmm::Filename, 584
- Trace
 - gdcmm::Trace, 1265
- TractographyResultsStorage
 - gdcmm::UIDs, 1309
- TransferSyntax
 - gdcmm::TransferSyntax, 1272
- TransferSyntaxArrayType
 - gdcmm::PresentationContext, 968
- TransferSyntaxes
 - gdcmm::PresentationContext, 970
- TransferSyntaxStringsType
 - gdcmm::UIDs, 1301
- TransferSyntaxSub
 - gdcmm::network::TransferSyntaxSub, 1276
- Transition
 - gdcmm::network::Transition, 1278
- transitions
 - gdcmm::network::TableRow, 1245
- TRIANGLE
 - gdcmm::MeshPrimitive, 815

- TRIANGLE_FAN
 - gdcm::MeshPrimitive, [815](#)
- TRIANGLE_STRIP
 - gdcm::MeshPrimitive, [815](#)
- Trim
 - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1193](#)
- TrimInternal
 - gdcm::CodeString, [315](#)
- Truncate
 - gdcm::String< TDelimiter, TMaxLength, TPadChar >, [1193](#)
- TryJPEG2000Codec
 - gdcm::Bitmap, [259](#)
 - gdcm::ImageChangeTransferSyntax, [655](#)
- TryJPEG2000Codec2
 - gdcm::Bitmap, [259](#)
- TryJPEGCodec
 - gdcm::Bitmap, [259](#)
 - gdcm::ImageChangeTransferSyntax, [655](#)
- TryJPEGCodec2
 - gdcm::Bitmap, [259](#)
- TryJPEGLSCodec
 - gdcm::Bitmap, [260](#)
 - gdcm::ImageChangeTransferSyntax, [655](#)
- TryKAKADUCodec
 - gdcm::Bitmap, [260](#)
- TryPVRGCodec
 - gdcm::Bitmap, [260](#)
- TryRAWCodec
 - gdcm::Bitmap, [260](#)
 - gdcm::ImageChangeTransferSyntax, [655](#)
- TryRLECodec
 - gdcm::Bitmap, [260](#)
 - gdcm::ImageChangeTransferSyntax, [655](#)
- TS
 - gdcm::Bitmap, [262](#)
- TS_END
 - gdcm::TransferSyntax, [1272](#)
- TSName
 - gdcm::UIDs, [1301](#)
- TSType
 - gdcm::TransferSyntax, [1271](#)
 - gdcm::UIDs, [1311](#)
- Type
 - gdcm::Element< TVR, TVM >, [458](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [465](#)
 - gdcm::Element< TVR, VM::VM1_n >, [469](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [477](#)
 - gdcm::Element< TVR, VM::VM2_n >, [483](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [489](#)
 - gdcm::Element< TVR, VM::VM3_4 >, [495](#)
 - gdcm::Element< TVR, VM::VM3_n >, [501](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [505](#)
 - gdcm::Element< VR::OB, VM::VM1 >, [510](#)
 - gdcm::Element< VR::OW, VM::VM1 >, [515](#)
 - gdcm::EquipmentManufacturer, [531](#)
 - gdcm::Type, [1280](#)
 - gdcm::VL, [1421](#)
- TYPETOENCODING
 - gdcmVR.h, [1829](#)
- TYPETOLENGTH
 - gdcmVM.h, [1825](#)
- TypeToString
 - gdcm::EquipmentManufacturer, [531](#)
- TypeType
 - gdcm::Type, [1280](#)
- UberonOntology
 - gdcm::UIDs, [1308](#)
- UC
 - gdcm::VR, [1433](#)
- UCComp
 - gdcm, [88](#)
- UI
 - gdcm::VR, [1433](#)
- UIComp
 - gdcm, [88](#)
- uid_1_2_840_10008_15_0_3_1
 - gdcm::UIDs, [1320](#)
- uid_1_2_840_10008_15_0_3_10
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_11
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_12
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_13
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_14
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_15
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_16
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_17
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_18
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_19
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_2
 - gdcm::UIDs, [1320](#)
- uid_1_2_840_10008_15_0_3_20
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_21
 - gdcm::UIDs, [1321](#)
- uid_1_2_840_10008_15_0_3_22
 - gdcm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_23
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_24
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_25
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_26
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_27
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_28
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_29
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_3
gdcmm::UIDs, [1320](#)

uid_1_2_840_10008_15_0_3_30
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_31
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_4
gdcmm::UIDs, [1320](#)

uid_1_2_840_10008_15_0_3_5
gdcmm::UIDs, [1320](#)

uid_1_2_840_10008_15_0_3_6
gdcmm::UIDs, [1320](#)

uid_1_2_840_10008_15_0_3_7
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_8
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_3_9
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_1
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_2
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_3
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_4
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_5
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_6
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_7
gdcmm::UIDs, [1321](#)

uid_1_2_840_10008_15_0_4_8
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_15_1_1
gdcmm::UIDs, [1327](#)

uid_1_2_840_10008_1_1
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_20
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_1_20_1
gdcmm::UIDs, [1313](#)

uid_1_2_840_10008_1_20_1_1
gdcmm::UIDs, [1313](#)

uid_1_2_840_10008_1_20_2
gdcmm::UIDs, [1313](#)

uid_1_2_840_10008_1_20_2_1
gdcmm::UIDs, [1313](#)

uid_1_2_840_10008_1_2_1
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_1_99
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_2
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_100
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_101
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_1_2_4_102
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_1_2_4_103
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_1_2_4_104
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_1_2_4_105
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_1_2_4_106
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_1_2_4_107
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_1_2_4_108
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_1_2_4_50
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_51
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_52
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_53
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_54
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_55
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_56
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_57
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_58
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_59
gdcmm::UIDs, [1312](#)

uid_1_2_840_10008_1_2_4_60
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_61
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_62
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_63
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_64
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_65
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_66
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_70
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_80
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_81
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_90
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_91
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_92
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_93
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_94
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_4_95
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_5
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_6_1
gdcmm::UIDs, [1312](#)
uid_1_2_840_10008_1_2_6_2
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_3_10
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_40
gdcmm::UIDs, [1314](#)
uid_1_2_840_10008_1_40_1
gdcmm::UIDs, [1314](#)
uid_1_2_840_10008_1_42
gdcmm::UIDs, [1314](#)
uid_1_2_840_10008_1_42_1
gdcmm::UIDs, [1314](#)
uid_1_2_840_10008_1_4_1_1
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_10
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_11
gdcmm::UIDs, [1313](#)

uid_1_2_840_10008_1_4_1_12
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_13
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_14
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_15
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_16
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_17
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_18
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_2
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_3
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_4
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_5
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_6
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_7
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_8
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_1_9
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_2_1
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_4_2_2
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_1_5_1
gdcmm::UIDs, [1323](#)
uid_1_2_840_10008_1_5_2
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_3
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_4
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_5
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_6
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_7
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_5_8
gdcmm::UIDs, [1322](#)
uid_1_2_840_10008_1_9
gdcmm::UIDs, [1313](#)
uid_1_2_840_10008_2_16_10
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_16_11
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_2_16_12
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_2_16_13
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_2_16_14
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_2_16_4
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_2_16_5
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_16_6
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_16_7
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_16_8
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_16_9
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_2_6_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_1_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_1_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_1_4
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_2_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_3_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_3_2
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_3_3
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_3_4
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_3_5
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_5_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_5_4
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_5_5
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_3_1_2_6_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_4_2
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_5_1_1_1
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_5_1_1_14
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_15
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_16
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_16_376
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_17
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_17_376
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_18
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_18_1
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_2
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_5_1_1_22
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_23
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_24
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_24_1
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_25
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_26
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_27
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_29
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_30
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_31
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_32
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_33
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_4
gdcmm::UIDs, [1314](#)

uid_1_2_840_10008_5_1_1_40
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_5_1_1_40_1
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_5_1_1_4_1
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_4_2
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_9
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_1_9_1
gdcmm::UIDs, [1315](#)

uid_1_2_840_10008_5_1_4_1_1_1
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_10
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_104_1
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_104_2
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_104_3
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_11
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_10
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_11
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_2
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_3
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_4
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_11_5
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_6
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_7
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_8
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_11_9
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_128
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_128_1
 gdcmm::UIDs, [1322](#)
 uid_1_2_840_10008_5_1_4_1_1_129
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_1_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_2_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_3
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_12_77
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_130
 gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_131
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_2
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_3
 gdcmm::UIDs, [1322](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_4
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_13_1_5
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_14_1
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_14_2
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_1_1
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1_1_1
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1_2
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1_2_1
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1_3
 gdcmm::UIDs, [1315](#)
 uid_1_2_840_10008_5_1_4_1_1_1_3_1
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_2
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_20
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_200_1
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_200_2
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_200_3
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_200_4
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_200_5
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_200_6
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_2_1
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_2_2
 gdcmm::UIDs, [1322](#)
 uid_1_2_840_10008_5_1_4_1_1_3
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_30
 gdcmm::UIDs, [1323](#)
 uid_1_2_840_10008_5_1_4_1_1_3_1
 gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_4
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_40
gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_5_1_4_1_1_481_1
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_10
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_481_11
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_481_2
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_3
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_4
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_5
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_6
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_7
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_8
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_481_9
gdcmm::UIDs, [1319](#)

uid_1_2_840_10008_5_1_4_1_1_4_1
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_4_2
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_4_3
gdcmm::UIDs, [1327](#)

uid_1_2_840_10008_5_1_4_1_1_4_4
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_5_1_4_1_1_5
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_501_1
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_2_1
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_2_2
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_3
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_4
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_5
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_501_6
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_6
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_601_1
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_601_2
gdcmm::UIDs, [1325](#)

uid_1_2_840_10008_5_1_4_1_1_66
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_66_1
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_66_2
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_66_3
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_66_4
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_66_5
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_5_1_4_1_1_66_6
gdcmm::UIDs, [1324](#)

uid_1_2_840_10008_5_1_4_1_1_67
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_68_1
gdcmm::UIDs, [1324](#)

uid_1_2_840_10008_5_1_4_1_1_68_2
gdcmm::UIDs, [1324](#)

uid_1_2_840_10008_5_1_4_1_1_6_1
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_6_2
gdcmm::UIDs, [1322](#)

uid_1_2_840_10008_5_1_4_1_1_7
gdcmm::UIDs, [1316](#)

uid_1_2_840_10008_5_1_4_1_1_77_1
gdcmm::UIDs, [1317](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcmm::UIDs, [1318](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_5
gdcmm::UIDs, [1324](#)

uid_1_2_840_10008_5_1_4_1_1_77_1_5_6
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_7
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_8
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_77_1_6
 gdcmm::UIDs, [1322](#)
 uid_1_2_840_10008_5_1_4_1_1_77_2
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_78_1
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_2
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_3
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_4
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_5
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_6
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_7
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_78_8
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_79_1
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_7_1
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_2
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_3
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_7_4
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_8
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_80_1
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_81_1
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_82_1
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_1
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_11
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_2
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_22
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_3
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_33
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_34
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_35
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_4
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_40
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_50
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_59
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_65
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_67
 gdcmm::UIDs, [1318](#)
 uid_1_2_840_10008_5_1_4_1_1_88_68
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_69
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_70
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_71
 gdcmm::UIDs, [1324](#)
 uid_1_2_840_10008_5_1_4_1_1_88_72
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_73
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_74
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_88_75
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_9
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_90_1
 gdcmm::UIDs, [1325](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_1
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_2
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_9_1_3
 gdcmm::UIDs, [1316](#)
 uid_1_2_840_10008_5_1_4_1_1_9_2_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_9_3_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_9_4_1
 gdcmm::UIDs, [1317](#)
 uid_1_2_840_10008_5_1_4_1_1_9_4_2
 gdcmm::UIDs, [1323](#)

uid_1_2_840_10008_5_1_4_1_1_9_5_1
gdcmm::UIDs, [1323](#)
uid_1_2_840_10008_5_1_4_1_1_9_6_1
gdcmm::UIDs, [1323](#)
uid_1_2_840_10008_5_1_4_1_2_1_1
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_1_2
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_1_3
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_2_1
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_2_2
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_2_3
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_3_1
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_3_2
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_3_3
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_1_2_4_2
gdcmm::UIDs, [1325](#)
uid_1_2_840_10008_5_1_4_1_2_4_3
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_1_2_5_3
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_20_1
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_20_2
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_20_3
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_31
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_32
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_32_1
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_32_2
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_32_3
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_33
gdcmm::UIDs, [1319](#)
uid_1_2_840_10008_5_1_4_34_1
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_10
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_2
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_3
gdcmm::UIDs, [1320](#)

uid_1_2_840_10008_5_1_4_34_4
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_4_1
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_4_2
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_4_3
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_4_4
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_5
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_34_5_1
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_6
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_6_1
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_6_2
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_6_3
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_6_4
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_7
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_8
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_34_9
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_37_1
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_37_2
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_37_3
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_38_1
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_38_2
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_38_3
gdcmm::UIDs, [1320](#)
uid_1_2_840_10008_5_1_4_38_4
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_39_1
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_39_2
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_39_3
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_39_4
gdcmm::UIDs, [1326](#)
uid_1_2_840_10008_5_1_4_41
gdcmm::UIDs, [1320](#)

- uid_1_2_840_10008_5_1_4_42
 - gdcm::UIDs, [1320](#)
- uid_1_2_840_10008_5_1_4_43_1
 - gdcm::UIDs, [1326](#)
- uid_1_2_840_10008_5_1_4_43_2
 - gdcm::UIDs, [1326](#)
- uid_1_2_840_10008_5_1_4_43_3
 - gdcm::UIDs, [1326](#)
- uid_1_2_840_10008_5_1_4_43_4
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_44_1
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_44_2
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_44_3
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_44_4
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_45_1
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_45_2
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_45_3
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_5_1_4_45_4
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_7_1_1
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_7_1_2
 - gdcm::UIDs, [1327](#)
- uid_1_2_840_10008_8_1_1
 - gdcm::UIDs, [1327](#)
- UIDGenerator
 - gdcm::UIDGenerator, [1283](#)
- UIDs
 - gdcm::UIDs, [1327](#)
- UIH
 - gdcm::EquipmentManufacturer, [531](#)
- UINT12
 - gdcm::PixelFormat, [932](#)
- UINT16
 - gdcm::PixelFormat, [932](#)
- UINT32
 - gdcm::PixelFormat, [932](#)
- UINT64
 - gdcm::PixelFormat, [932](#)
- UINT8
 - gdcm::PixelFormat, [932](#)
- UL
 - gdcm::VR, [1433](#)
- ULAction
 - gdcm::network::ULAction, [1331](#)
- ULActionAE6
 - gdcm::network::ULConnection, [1377](#)
- ULBasicCallback
 - gdcm::network::ULBasicCallback, [1372](#)
- ULConnection
 - gdcm::network::ULConnection, [1374](#)
- ULConnectionCallback
 - gdcm::network::ULConnectionCallback, [1379](#)
- ULConnectionInfo
 - gdcm::network::ULConnectionInfo, [1381](#)
- ULConnectionManager
 - gdcm::network::ULConnection, [1377](#)
 - gdcm::network::ULConnectionManager, [1385](#)
- ULError
 - gdcm::network::ULError, [1390](#)
- ULTransitionTable
 - gdcm::network::ULTransitionTable, [1392](#)
- UltrasoundImageStorage
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- UltrasoundImageStorageRetired
 - gdcm::MediaStorage, [799](#)
 - gdcm::UIDs, [1304](#)
- UltrasoundMultiFrameImageStorage
 - gdcm::MediaStorage, [799](#)
- UltrasoundMultiframeImageStorage
 - gdcm::UIDs, [1304](#)
- UltrasoundMultiFrameImageStorageRetired
 - gdcm::MediaStorage, [799](#)
- UltrasoundMultiframeImageStorageRetired
 - gdcm::UIDs, [1304](#)
- ULWritingCallback
 - gdcm::network::ULWritingCallback, [1394](#)
- UN
 - gdcm::VR, [1433](#)
- UndefinedEntityError
 - gdcm::Parser, [902](#)
- underline
 - gdcm::terminal, [110](#)
- UnexpectedStateError
 - gdcm::Parser, [902](#)
- UnifiedProcedureStepEventSOPClass
 - gdcm::UIDs, [1306](#)
- UnifiedProcedureStepEventSOPClass1
 - gdcm::UIDs, [1311](#)
- UnifiedProcedureStepPullSOPClass
 - gdcm::UIDs, [1306](#)
- UnifiedProcedureStepPullSOPClass1
 - gdcm::UIDs, [1311](#)
- UnifiedProcedureStepPushSOPClass
 - gdcm::UIDs, [1306](#)
- UnifiedProcedureStepPushSOPClass1
 - gdcm::UIDs, [1311](#)
- UnifiedProcedureStepWatchSOPClass
 - gdcm::UIDs, [1306](#)
- UnifiedProcedureStepWatchSOPClass1

- gdcm::UIDs, [1311](#)
- UnifiedWorklistandProcedureStepServiceClass
 - gdcm::UIDs, [1306](#)
- UnifiedWorklistandProcedureStepServiceClass1
 - gdcm::UIDs, [1311](#)
- UnifiedWorklistandProcedureStepSOPInstance
 - gdcm::UIDs, [1306](#)
- UnInstallPipeline
 - vtkImageColorViewer, [1519](#)
- UniversalCoordinatedTime
 - gdcm::UIDs, [1311](#)
- UNKNOWN
 - gdcm::CSAHeader, [350](#)
 - gdcm::EquipmentManufacturer, [531](#)
 - gdcm::LookupTable, [779](#)
 - gdcm::Orientation, [887](#)
 - gdcm::PhotometricInterpretation, [927](#)
 - gdcm::PixelFormat, [933](#)
 - gdcm::Spacing, [1150](#)
 - gdcm::Surface, [1205](#)
 - gdcm::Type, [1280](#)
- Unknown
 - gdcm::SwapCode, [1226](#)
 - gdcm::TransferSyntax, [1271](#)
- Unpack
 - gdcm::Unpacker12Bits, [1403](#)
- UnRegister
 - gdcm::Object, [874](#)
- UnusedBitsPresentInPixelData
 - gdcm::Bitmap, [260](#)
 - gdcm::Pixmap, [945](#)
- Update
 - gdcm::Curve, [369](#)
 - gdcm::Overlay, [898](#)
- UpdateDisplayExtent
 - vtkImageColorViewer, [1519](#)
- UpdateOrientation
 - vtkImageColorViewer, [1519](#)
- UpdatePosition
 - gdcm::ByteBuffer, [271](#)
- UPSFilteredGlobalSubscriptionSOPInstance
 - gdcm::UIDs, [1311](#)
- UR
 - gdcm::VR, [1433](#)
- URComp
 - gdcm, [88](#)
- URI
 - gdcm::MediaStorage, [802](#)
- US
 - gdcm::VR, [1433](#)
- US_OW
 - gdcm::VR, [1433](#)
- US_SS
 - gdcm::VR, [1433](#)
- US_SS_OW
 - gdcm::VR, [1433](#)
- Usage
 - gdcm::Usage, [1405](#)
- UsageType
 - gdcm::Usage, [1405](#)
- UseDictAlways
 - gdcm::PythonFilter, [1005](#)
 - gdcm::StringFilter, [1197](#)
- UseGrayscaleSecondaryImageStorage
 - gdcm::EmptyMaskGenerator, [521](#)
- UseOriginalSOPClassUID
 - gdcm::EmptyMaskGenerator, [521](#)
- UserInformation
 - gdcm::network::UserInformation, [1409](#)
- UserOption
 - gdcm::Usage, [1405](#)
- UserOrdering
 - gdcm::SerieHelper, [1113](#)
- UT
 - gdcm::VR, [1433](#)
- UTComp
 - gdcm, [88](#)
- Utilities Directory Reference, [68](#)
- UV
 - gdcm::VR, [1433](#)
- V
 - gdcm::Validate, [1413](#)
- Valid
 - gdcm::Preamble, [966](#)
- Validate
 - gdcm::PixelFormat, [938](#)
 - gdcm::Validate, [1412](#)
- ValidateQuery
 - gdcm::BaseQuery, [235](#)
 - gdcm::BaseRootQuery, [239](#)
 - gdcm::FindPatientRootQuery, [605](#)
 - gdcm::FindStudyRootQuery, [609](#)
 - gdcm::ModalityPerformedProcedureStepCreate-Query, [820](#)
 - gdcm::ModalityPerformedProcedureStepSet-Query, [824](#)
 - gdcm::MovePatientRootQuery, [839](#)
 - gdcm::MoveStudyRootQuery, [843](#)
 - gdcm::WLMFindQuery, [1557](#)
- Validation
 - gdcm::Validate, [1413](#)
- ValidDataSet
 - gdcm::BaseQuery, [235](#)
- Value
 - gdcm::Value, [1415](#)
- value

- gdcmm::STATIC_ASSERTION_FAILURE< true
 >, [1158](#)
- value_type
 - gdcmm::CodeString, [314](#)
 - gdcmm::LO, [776](#)
 - gdcmm::String< TDelimiter, TMaxLength, TPad-
 Char >, [1191](#)
- ValueField
 - gdcmm::DataElement, [383](#)
 - gdcmm::PDBelement, [910](#)
- ValueLengthField
 - gdcmm::DataElement, [383](#)
- ValueMultiplicityField
 - gdcmm::CSAElement, [348](#)
- ValuePtr
 - gdcmm::DataElement, [373](#)
- ValueType
 - gdcmm::Scanner, [1053](#)
 - gdcmm::Scanner2, [1063](#)
 - gdcmm::StrictScanner, [1173](#)
 - gdcmm::StrictScanner2, [1182](#)
- VERBOSE_STYLE
 - gdcmm::Printer, [985](#)
- VerificationSOPClass
 - gdcmm::UIDs, [1301](#)
- Verify
 - gdcmm::Defs, [410](#), [411](#)
 - gdcmm::Macro, [789](#)
 - gdcmm::Module, [828](#)
- Version
 - gdcmm::Version, [1419](#)
- VERTEX
 - gdcmm::MeshPrimitive, [815](#)
- Video
 - gdcmm::MediaStorage, [802](#)
- VideoEndoscopicImageStorage
 - gdcmm::MediaStorage, [800](#)
 - gdcmm::UIDs, [1305](#)
- VideoMicroscopicImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- VideoPhotographicImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- VIEWType
 - gdcmm::Surface, [1205](#)
- VIEWType_END
 - gdcmm::Surface, [1205](#)
- VisualAcuityMeasurementsStorage
 - gdcmm::UIDs, [1309](#)
- VL
 - gdcmm::VL, [1422](#)
- VL16
 - gdcmm::VR, [1433](#)
- VL32
 - gdcmm::VR, [1433](#)
- VLEndoscopicImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- VLIImageStorageTrialRetired
 - gdcmm::UIDs, [1305](#)
- VLMicroscopicImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1305](#)
- VLMultiframeImageStorageTrialRetired
 - gdcmm::UIDs, [1305](#)
- VLPhotographicImageStorage
 - gdcmm::MediaStorage, [800](#)
 - gdcmm::UIDs, [1305](#)
- VLSlideCoordinatesMicroscopicImageStorage
 - gdcmm::UIDs, [1305](#)
- VLWholeSlideMicroscopyImageStorage
 - gdcmm::MediaStorage, [801](#)
 - gdcmm::UIDs, [1308](#)
- VM
 - gdcmm::VM, [1428](#)
- VM0
 - gdcmm::VM, [1426](#)
- VM1
 - gdcmm::VM, [1426](#)
- VM10
 - gdcmm::VM, [1427](#)
- VM12
 - gdcmm::VM, [1427](#)
- VM16
 - gdcmm::VM, [1427](#)
- VM18
 - gdcmm::VM, [1427](#)
- VM1_2
 - gdcmm::VM, [1427](#)
- VM1_3
 - gdcmm::VM, [1427](#)
- VM1_32
 - gdcmm::VM, [1427](#)
- VM1_4
 - gdcmm::VM, [1427](#)
- VM1_5
 - gdcmm::VM, [1427](#)
- VM1_8
 - gdcmm::VM, [1427](#)
- VM1_99
 - gdcmm::VM, [1427](#)
- VM1_n
 - gdcmm::VM, [1427](#)
- VM2
 - gdcmm::VM, [1426](#)
- VM24
 - gdcmm::VM, [1427](#)

- VM256
 - gdcm::VM, [1427](#)
- VM28
 - gdcm::VM, [1427](#)
- VM2_2n
 - gdcm::VM, [1427](#)
- VM2_n
 - gdcm::VM, [1427](#)
- VM3
 - gdcm::VM, [1426](#)
- VM30_30n
 - gdcm::VM, [1427](#)
- VM32
 - gdcm::VM, [1427](#)
- VM35
 - gdcm::VM, [1427](#)
- VM3_3n
 - gdcm::VM, [1427](#)
- VM3_4
 - gdcm::VM, [1427](#)
- VM3_n
 - gdcm::VM, [1427](#)
- VM4
 - gdcm::VM, [1426](#)
- VM47_47n
 - gdcm::VM, [1427](#)
- VM4_4n
 - gdcm::VM, [1427](#)
- VM5
 - gdcm::VM, [1426](#)
- VM6
 - gdcm::VM, [1427](#)
- VM6_6n
 - gdcm::VM, [1427](#)
- VM6_n
 - gdcm::VM, [1427](#)
- VM7_7n
 - gdcm::VM, [1427](#)
- VM8
 - gdcm::VM, [1427](#)
- VM9
 - gdcm::VM, [1427](#)
- VM99
 - gdcm::VM, [1427](#)
- VM__END
 - gdcm::VM, [1428](#)
- VMType
 - gdcm::Attribute< Group, Element, TVR, TVM >, [161](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [171](#)
 - gdcm::VM, [1426](#)
- VOILUTBoxSOPClass
 - gdcm::UIDs, [1303](#)
- VolumeRenderingVolumetricPresentationStateStorage
 - gdcm::UIDs, [1309](#)
- VR
 - gdcm::VR, [1433](#)
- VR__END
 - gdcm::VR, [1433](#)
- VR__VM1
 - gdcm::VR, [1433](#)
- VRALL
 - gdcm::VR, [1433](#)
- VRASCII
 - gdcm::VR, [1433](#)
- VRBINARY
 - gdcm::VR, [1433](#)
- VRField
 - gdcm::CSAElement, [348](#)
 - gdcm::DataElement, [383](#)
- VRType
 - gdcm::VR, [1432](#)
- VRTypeTemplateCase
 - gdcmVR.h, [1829](#)
- VT100
 - gdcm::terminal, [110](#)
- VTK Directory Reference, [68](#)
- VTK__CMYK
 - vtkGDCMImageReader.h, [2222](#)
 - vtkGDCMImageReader2.h, [2227](#)
- VTK__INVERSE__LUMINANCE
 - vtkGDCMImageReader.h, [2222](#)
 - vtkGDCMImageReader2.h, [2227](#)
- VTK__LOOKUP__TABLE
 - vtkGDCMImageReader.h, [2222](#)
 - vtkGDCMImageReader2.h, [2227](#)
- VTK__YBR
 - vtkGDCMImageReader.h, [2222](#)
 - vtkGDCMImageReader2.h, [2227](#)
- vtkBooleanMacro
 - vtkGDCMImageReader, [1451](#)
 - vtkGDCMImageReader2, [1466](#)
 - vtkGDCMImageWriter, [1478](#)
 - vtkGDCMThreadedImageReader, [1502](#)
 - vtkGDCMThreadedImageReader2, [1506](#)
 - vtkImageColorViewer, [1520](#)
 - vtkImageMapToColors16, [1526](#)
- vtkGDCMImageReader, [1445](#)
- ~vtkGDCMImageReader, [1448](#)
- ApplyInverseVideo, [1456](#)
- ApplyLookupTable, [1456](#)
- ApplyPlanarConfiguration, [1456](#)
- ApplyShiftScale, [1456](#)
- ApplyYBRToRGB, [1457](#)
- CanReadFile, [1448](#)
- Curve, [1457](#)

- DirectionCosines, [1457](#)
- ExecuteData, [1448](#)
- ExecuteInformation, [1448](#)
- FileNames, [1457](#)
- FillMedicalImageInformation, [1448](#)
- ForceRescale, [1457](#)
- GetDescriptiveName, [1449](#)
- GetFileExtensions, [1449](#)
- GetIconImage, [1449](#)
- GetOverlay, [1449](#)
- IconDataScalarType, [1457](#)
- IconImageDataExtent, [1457](#)
- IconNumberOfScalarComponents, [1457](#)
- ImageFormat, [1458](#)
- ImageOrientationPatient, [1458](#)
- ImagePositionPatient, [1458](#)
- LoadIconImage, [1458](#)
- LoadOverlays, [1458](#)
- LoadSingleFile, [1449](#)
- LossyFlag, [1458](#)
- MedicalImageProperties, [1458](#)
- New, [1449](#)
- NumberOfIconImages, [1459](#)
- NumberOfOverlays, [1459](#)
- PlanarConfiguration, [1459](#)
- PrintSelf, [1449](#)
- RequestDataCompat, [1450](#)
- RequestInformationCompat, [1450](#)
- Scale, [1459](#)
- SetCurve, [1450](#)
- SetFileNames, [1450](#)
- SetFilePattern, [1450](#)
- SetFilePrefix, [1451](#)
- SetMedicalImageProperties, [1451](#)
- Shift, [1459](#)
- vtkBooleanMacro, [1451](#)
- vtkGDCMImageReader, [1448](#)
- vtkGDCMMedicalImageProperties, [1484](#)
- vtkGetMacro, [1452](#), [1453](#)
- vtkGetObjectMacro, [1453](#), [1454](#)
- vtkGetStringMacro, [1454](#)
- vtkGetVector3Macro, [1454](#)
- vtkGetVector6Macro, [1455](#)
- vtkSetMacro, [1455](#)
- vtkSetVector6Macro, [1456](#)
- vtkTypeMacro, [1456](#)
- vtkGDCMImageReader.h, [2220](#), [2222](#)
 - VTK_CMYK, [2222](#)
 - VTK_INVERSE_LUMINANCE, [2222](#)
 - VTK_LOOKUP_TABLE, [2222](#)
 - VTK_YBR, [2222](#)
- vtkGDCMImageReader2, [1460](#)
 - ~vtkGDCMImageReader2, [1462](#)
 - ApplyInverseVideo, [1471](#)
 - ApplyLookupTable, [1471](#)
 - ApplyPlanarConfiguration, [1471](#)
 - ApplyShiftScale, [1471](#)
 - ApplyYBRToRGB, [1471](#)
 - CanReadFile, [1463](#)
 - Curve, [1471](#)
 - DirectionCosines, [1471](#)
 - FillMedicalImageInformation, [1463](#)
 - ForceRescale, [1471](#)
 - GetDescriptiveName, [1463](#)
 - GetFileExtensions, [1463](#)
 - GetIconImage, [1463](#)
 - GetIconImagePort, [1463](#)
 - GetOverlay, [1463](#)
 - GetOverlayPort, [1463](#)
 - IconDataScalarType, [1471](#)
 - IconImageDataExtent, [1472](#)
 - IconNumberOfScalarComponents, [1472](#)
 - ImageFormat, [1472](#)
 - ImageOrientationPatient, [1472](#)
 - ImagePositionPatient, [1472](#)
 - LoadIconImage, [1472](#)
 - LoadOverlays, [1472](#)
 - LoadSingleFile, [1463](#)
 - LossyFlag, [1472](#)
 - New, [1464](#)
 - NumberOfIconImages, [1473](#)
 - NumberOfOverlays, [1473](#)
 - PlanarConfiguration, [1473](#)
 - PrintSelf, [1464](#)
 - ProcessRequest, [1464](#)
 - RequestData, [1464](#)
 - RequestDataCompat, [1464](#)
 - RequestInformation, [1465](#)
 - RequestInformationCompat, [1465](#)
 - Scale, [1473](#)
 - SetCurve, [1465](#)
 - SetFilePattern, [1465](#)
 - SetFilePrefix, [1465](#)
 - SetMedicalImageProperties, [1465](#)
 - Shift, [1473](#)
 - vtkBooleanMacro, [1466](#)
 - vtkGDCMImageReader2, [1462](#)
 - vtkGDCMMedicalImageProperties, [1484](#)
 - vtkGetMacro, [1466–1468](#)
 - vtkGetObjectMacro, [1468](#)
 - vtkGetStringMacro, [1469](#)
 - vtkGetVector3Macro, [1469](#)
 - vtkGetVector6Macro, [1469](#)
 - vtkSetMacro, [1469](#), [1470](#)
 - vtkSetVector6Macro, [1470](#)
 - vtkTypeMacro, [1470](#)
- vtkGDCMImageReader2.h, [2226](#), [2228](#)
 - VTK_CMYK, [2227](#)

- VTK_INVERSE_LUMINANCE, 2227
- VTK_LOOKUP_TABLE, 2227
- VTK_YBR, 2227
- vtkGDCMImageWriter, 1474
 - ~vtkGDCMImageWriter, 1476
 - CompressionTypes, 1476
 - GetDescriptiveName, 1476
 - GetFileExtensions, 1476
 - GetFileName, 1477
 - JPEG2000_COMPRESSION, 1476
 - JPEG_COMPRESSION, 1476
 - JPEGLS_COMPRESSION, 1476
 - New, 1477
 - NO_COMPRESSION, 1476
 - PrintSelf, 1477
 - RLE_COMPRESSION, 1476
 - SetDirectionCosines, 1477
 - SetDirectionCosinesFromImageOrientationPa-
tient, 1477
 - SetFileNames, 1477
 - SetMedicalImageProperties, 1478
 - vtkBooleanMacro, 1478
 - vtkGDCMImageWriter, 1476
 - vtkGDCMMedicalImageProperties, 1485
 - vtkGetMacro, 1478, 1479
 - vtkGetObjectMacro, 1479
 - vtkGetStringMacro, 1480
 - vtkSetMacro, 1480, 1481
 - vtkSetStringMacro, 1481
 - vtkTypeMacro, 1481
 - Write, 1481
 - WriteGDCMData, 1481
 - WriteSlice, 1482
- vtkGDCMImageWriter.h, 2231, 2232
- vtkGDCMMedicalImageProperties, 1482
 - ~vtkGDCMMedicalImageProperties, 1483
 - Clear, 1483
 - GetFile, 1483
 - New, 1484
 - PrintSelf, 1484
 - PushBackFile, 1484
 - vtkGDCMImageReader, 1484
 - vtkGDCMImageReader2, 1484
 - vtkGDCMImageWriter, 1485
 - vtkGDCMMedicalImageProperties, 1483
 - vtkTypeMacro, 1484
- vtkGDCMMedicalImageProperties.h, 2234, 2235
- vtkGDCMPolyDataReader, 1485
 - ~vtkGDCMPolyDataReader, 1487
 - FileName, 1489
 - FillMedicalImageInformation, 1487
 - MedicalImageProperties, 1489
 - New, 1487
 - PrintSelf, 1487
- RequestData, 1488
- RequestData_HemodynamicWaveformStorage, 1488
- RequestData_RTStructureSetStorage, 1488
- RequestInformation, 1488
- RequestInformation_HemodynamicWaveform-
Storage, 1488
- RequestInformation_RTStructureSetStorage, 1488
- RTStructSetProperties, 1490
- vtkGDCMPolyDataReader, 1487
- vtkGetObjectMacro, 1488, 1489
- vtkGetStringMacro, 1489
- vtkSetStringMacro, 1489
- vtkTypeMacro, 1489
- vtkGDCMPolyDataReader.h, 2240
- vtkGDCMPolyDataWriter, 1490
 - ~vtkGDCMPolyDataWriter, 1492
 - InitializeRTStructSet, 1492
 - MedicalImageProperties, 1494
 - New, 1492
 - PrintSelf, 1492
 - RTStructSetProperties, 1494
 - SetMedicalImageProperties, 1493
 - SetNumberOfInputPorts, 1493
 - SetRTStructSetProperties, 1493
 - vtkGDCMPolyDataWriter, 1492
 - vtkTypeMacro, 1493
 - WriteData, 1493
 - WriteRTSTRUCTData, 1493
 - WriteRTSTRUCTInfo, 1494
- vtkGDCMPolyDataWriter.h, 2241, 2242
- vtkGDCMTesting, 1494
 - ~vtkGDCMTesting, 1496
 - GetGDCMDataRoot, 1496
 - GetMD5MetaImage, 1496
 - GetMHDMD5FromFile, 1496
 - GetNumberOfMD5MetaImages, 1496
 - GetRAWMD5FromFile, 1497
 - GetVTKDataRoot, 1497
 - MD5MetaImagesType, 1496
 - New, 1497
 - PrintSelf, 1497
 - vtkGDCMTesting, 1496
 - vtkTypeMacro, 1497
- vtkGDCMTesting.h, 2243, 2244
- vtkGDCMThreadedImageReader, 1498
 - ~vtkGDCMThreadedImageReader, 1501
 - ExecuteData, 1501
 - ExecuteInformation, 1501
 - New, 1501
 - PrintSelf, 1501
 - ReadFiles, 1502
 - RequestDataCompat, 1502

- vtkBooleanMacro, 1502
 - vtkGDCMThreadedImageReader, 1501
 - vtkGetMacro, 1502
 - vtkSetMacro, 1502
 - vtkTypeMacro, 1503
- vtkGDCMThreadedImageReader.h, 2245
- vtkGDCMThreadedImageReader2, 1503
 - ~vtkGDCMThreadedImageReader2, 1505
 - GetFileName, 1505
 - New, 1505
 - PrintSelf, 1505
 - RequestInformation, 1505
 - SetFileName, 1505
 - SetFileNames, 1505
 - SplitExtent, 1506
 - ThreadedRequestData, 1506
- vtkBooleanMacro, 1506
- vtkGDCMThreadedImageReader2, 1505
- vtkGetMacro, 1506, 1507
- vtkGetObjectMacro, 1508
- vtkGetVector3Macro, 1508
- vtkGetVector6Macro, 1508
- vtkSetMacro, 1508, 1509
- vtkSetVector3Macro, 1509
- vtkSetVector6Macro, 1509
- vtkTypeMacro, 1510
- vtkGDCMThreadedImageReader2.h, 2247
- vtkGetMacro
 - vtkGDCMImageReader, 1452, 1453
 - vtkGDCMImageReader2, 1466–1468
 - vtkGDCMImageWriter, 1478, 1479
 - vtkGDCMThreadedImageReader, 1502
 - vtkGDCMThreadedImageReader2, 1506, 1507
 - vtkImageColorViewer, 1520
 - vtkImageMapToColors16, 1526, 1527
 - vtkImageMapToWindowLevelColors2, 1531
- vtkGetObjectMacro
 - vtkGDCMImageReader, 1453, 1454
 - vtkGDCMImageReader2, 1468
 - vtkGDCMImageWriter, 1479
 - vtkGDCMPolyDataReader, 1488, 1489
 - vtkGDCMThreadedImageReader2, 1508
 - vtkImageColorViewer, 1520, 1521
 - vtkImageMapToColors16, 1527
- vtkGetStringMacro
 - vtkGDCMImageReader, 1454
 - vtkGDCMImageReader2, 1469
 - vtkGDCMImageWriter, 1480
 - vtkGDCMPolyDataReader, 1489
 - vtkRTStructSetProperties, 1548–1550
- vtkGetVector3Macro
 - vtkGDCMImageReader, 1454
 - vtkGDCMImageReader2, 1469
 - vtkGDCMThreadedImageReader2, 1508
- vtkGetVector6Macro
 - vtkGDCMImageReader, 1455
 - vtkGDCMImageReader2, 1469
 - vtkGDCMThreadedImageReader2, 1508
- vtkImageColorViewer, 1510
 - ~vtkImageColorViewer, 1513
 - AddInput, 1514
 - AddInputConnection, 1514
 - FirstRender, 1521
 - GetColorLevel, 1514
 - GetColorWindow, 1514
 - GetInput, 1514
 - GetOffScreenRendering, 1514
 - GetOverlayVisibility, 1514
 - GetPosition, 1514
 - GetSize, 1514
 - GetSliceMax, 1515
 - GetSliceMin, 1515
 - GetSliceRange, 1515
 - GetWindowName, 1515
 - ImageActor, 1521
 - InstallPipeline, 1515
 - Interactor, 1522
 - InteractorStyle, 1522
 - New, 1515
 - OverlayImageActor, 1522
 - PrintSelf, 1516
 - Render, 1516
 - Renderer, 1522
 - RenderWindow, 1522
 - SetColorLevel, 1516
 - SetColorWindow, 1516
 - SetDisplayId, 1516
 - SetInput, 1516
 - SetInputConnection, 1517
 - SetOffScreenRendering, 1517
 - SetOverlayVisibility, 1517
 - SetParentId, 1517
 - SetPosition, 1517
 - SetRenderer, 1517
 - SetRenderWindow, 1518
 - SetSize, 1518
 - SetSlice, 1518
 - SetSliceOrientation, 1518
 - SetSliceOrientationToXY, 1518
 - SetSliceOrientationToXZ, 1519
 - SetSliceOrientationToYZ, 1519
 - SetupInteractor, 1519
 - SetWindowId, 1519
 - Slice, 1522
 - SLICE_ORIENTATION_XY, 1513
 - SLICE_ORIENTATION_XZ, 1513
 - SLICE_ORIENTATION_YZ, 1513
 - SliceOrientation, 1522

- UnInstallPipeline, [1519](#)
- UpdateDisplayExtent, [1519](#)
- UpdateOrientation, [1519](#)
- vtkBooleanMacro, [1520](#)
- vtkGetMacro, [1520](#)
- vtkGetObjectMacro, [1520](#), [1521](#)
- vtkImageColorViewer, [1513](#)
- vtkImageColorViewerCallback, [1521](#)
- vtkTypeMacro, [1521](#)
- WindowLevel, [1522](#)
- vtkImageColorViewer.h, [2249](#), [2250](#)
- vtkImageColorViewerCallback
 - vtkImageColorViewer, [1521](#)
- vtkImageMapToColors16, [1523](#)
 - ~vtkImageMapToColors16, [1524](#)
 - ActiveComponent, [1528](#)
 - DataWasPassed, [1528](#)
 - GetMTime, [1525](#)
 - LookupTable, [1528](#)
 - New, [1525](#)
 - OutputFormat, [1528](#)
 - PassAlphaToOutput, [1528](#)
 - PrintSelf, [1525](#)
 - RequestData, [1525](#)
 - RequestInformation, [1525](#)
 - SetLookupTable, [1525](#)
 - SetOutputFormatToLuminance, [1525](#)
 - SetOutputFormatToLuminanceAlpha, [1526](#)
 - SetOutputFormatToRGB, [1526](#)
 - SetOutputFormatToRGBA, [1526](#)
 - ThreadedRequestData, [1526](#)
 - vtkBooleanMacro, [1526](#)
 - vtkGetMacro, [1526](#), [1527](#)
 - vtkGetObjectMacro, [1527](#)
 - vtkImageMapToColors16, [1524](#)
 - vtkSetMacro, [1527](#)
 - vtkTypeMacro, [1528](#)
- vtkImageMapToColors16.h, [2253](#), [2254](#)
- vtkImageMapToWindowLevelColors2, [1529](#)
 - ~vtkImageMapToWindowLevelColors2, [1530](#)
 - Level, [1532](#)
 - New, [1531](#)
 - PrintSelf, [1531](#)
 - RequestData, [1531](#)
 - RequestInformation, [1531](#)
 - ThreadedRequestData, [1531](#)
 - vtkGetMacro, [1531](#)
 - vtkImageMapToWindowLevelColors2, [1530](#)
 - vtkSetMacro, [1532](#)
 - vtkTypeMacro, [1532](#)
 - Window, [1532](#)
- vtkImageMapToWindowLevelColors2.h, [2255](#), [2256](#)
- vtkImagePlanarComponentsToComponents, [1533](#)
 - ~vtkImagePlanarComponentsToComponents, [1534](#)
 - New, [1534](#)
 - PrintSelf, [1534](#)
 - RequestData, [1534](#)
 - vtkImagePlanarComponentsToComponents, [1534](#)
 - vtkTypeMacro, [1534](#)
- vtkImagePlanarComponentsToComponents.h, [2257](#), [2258](#)
- vtkImageRGBToYBR, [1535](#)
 - ~vtkImageRGBToYBR, [1536](#)
 - New, [1536](#)
 - PrintSelf, [1536](#)
 - ThreadedExecute, [1536](#)
 - vtkImageRGBToYBR, [1536](#)
 - vtkTypeMacro, [1537](#)
- vtkImageRGBToYBR.h, [2259](#), [2260](#)
- vtkImageYBRToRGB, [1537](#)
 - ~vtkImageYBRToRGB, [1538](#)
 - New, [1539](#)
 - PrintSelf, [1539](#)
 - ThreadedExecute, [1539](#)
 - vtkImageYBRToRGB, [1538](#)
 - vtkTypeMacro, [1539](#)
- vtkImageYBRToRGB.h, [2261](#)
- vtkLookupTable16, [1540](#)
 - ~vtkLookupTable16, [1541](#)
 - Build, [1541](#)
 - GetPointer, [1541](#)
 - MapScalarsThroughTable2, [1541](#)
 - New, [1542](#)
 - PrintSelf, [1542](#)
 - SetNumberOfTableValues, [1542](#)
 - Table16, [1543](#)
 - vtkLookupTable16, [1541](#)
 - vtkTypeMacro, [1542](#)
 - WritePointer, [1542](#)
- vtkLookupTable16.h, [2262](#), [2263](#)
- vtkRTStructSetProperties, [1543](#)
 - ~vtkRTStructSetProperties, [1545](#)
 - AddContourReferencedFrameOfReference, [1545](#)
 - AddReferencedFrameOfReference, [1545](#)
 - AddStructureSetROI, [1546](#)
 - AddStructureSetROIObservation, [1546](#)
 - Clear, [1546](#)
 - DeepCopy, [1546](#)
 - GetContourReferencedFrameOfReferenceClassUID, [1546](#)
 - GetContourReferencedFrameOfReferenceInstanceUID, [1546](#)
 - GetNumberOfContourReferencedFrameOfReferences, [1546](#), [1547](#)

- GetNumberOfReferencedFrameOfReferences, 1547
- GetNumberOfStructureSetROIs, 1547
- GetReferencedFrameOfReferenceClassUID, 1547
- GetReferencedFrameOfReferenceInstanceUID, 1547
- GetStructureSetObservationNumber, 1547
- GetStructureSetROIDescription, 1547
- GetStructureSetROIGenerationAlgorithm, 1547
- GetStructureSetROIName, 1547
- GetStructureSetROINumber, 1548
- GetStructureSetROIObservationLabel, 1548
- GetStructureSetROIRefFrameRefUID, 1548
- GetStructureSetRTROIInterpretedType, 1548
- Internals, 1552
- New, 1548
- PrintSelf, 1548
- ReferenceFrameOfReferenceUID, 1552
- ReferenceSeriesInstanceUID, 1552
- SeriesInstanceUID, 1552
- SOPInstanceUID, 1552
- StructureSetDate, 1552
- StructureSetLabel, 1552
- StructureSetName, 1553
- StructureSetTime, 1553
- StudyInstanceUID, 1553
- vtkGetStringMacro, 1548–1550
- vtkRTStructSetProperties, 1545
- vtkSetStringMacro, 1550, 1551
- vtkTypeMacro, 1551
- vtkRTStructSetProperties.h, 2264
- vtkSetMacro
 - vtkGDCMImageReader, 1455
 - vtkGDCMImageReader2, 1469, 1470
 - vtkGDCMImageWriter, 1480, 1481
 - vtkGDCMThreadedImageReader, 1502
 - vtkGDCMThreadedImageReader2, 1508, 1509
 - vtkImageMapToColors16, 1527
 - vtkImageMapToWindowLevelColors2, 1532
- vtkSetStringMacro
 - vtkGDCMImageWriter, 1481
 - vtkGDCMPolyDataReader, 1489
 - vtkRTStructSetProperties, 1550, 1551
- vtkSetVector3Macro
 - vtkGDCMThreadedImageReader2, 1509
- vtkSetVector6Macro
 - vtkGDCMImageReader, 1456
 - vtkGDCMImageReader2, 1470
 - vtkGDCMThreadedImageReader2, 1509
- vtkTypeMacro
 - vtkGDCMImageReader, 1456
 - vtkGDCMImageReader2, 1470
 - vtkGDCMImageWriter, 1481
 - vtkGDCMMedicalImageProperties, 1484
 - vtkGDCMPolyDataReader, 1489
 - vtkGDCMPolyDataWriter, 1493
 - vtkGDCMTesting, 1497
 - vtkGDCMThreadedImageReader, 1503
 - vtkGDCMThreadedImageReader2, 1510
 - vtkImageColorViewer, 1521
 - vtkImageMapToColors16, 1528
 - vtkImageMapToWindowLevelColors2, 1532
 - vtkImagePlanarComponentsToComponents, 1534
 - vtkImageRGBToYBR, 1537
 - vtkImageYBRToRGB, 1539
 - vtkLookupTable16, 1542
 - vtkRTStructSetProperties, 1551
- WarningOff
 - gdcmm::Trace, 1268
- WarningOn
 - gdcmm::Trace, 1268
- Waveform
 - gdcmm::MediaStorage, 802
 - gdcmm::Waveform, 1554
- WaveformStorageTrialRetired
 - gdcmm::UIDs, 1304
- WeirdPapryus
 - gdcmm::TransferSyntax, 1272
- what
 - gdcmm::Exception, 537
- white
 - gdcmm::terminal, 110
- WideFieldOphthalmicPhotography3DCoordinates-ImageStorage
 - gdcmm::UIDs, 1309
- WideFieldOphthalmicPhotographyStereographicProjectionImageStorage
 - gdcmm::UIDs, 1309
- Window
 - vtkImageMapToWindowLevelColors2, 1532
- WindowLevel
 - vtkImageColorViewer, 1522
- WinterColorPaletteSOPInstance
 - gdcmm::UIDs, 1308
- WIREFRAME
 - gdcmm::Surface, 1205
- WLMFindQuery
 - gdcmm::WLMFindQuery, 1557
- Wrapping Directory Reference, 69
- Write
 - gdcmm::ByteValue, 283
 - gdcmm::CommandDataSet, 323
 - gdcmm::DataElement, 382
 - gdcmm::DataSet, 399
 - gdcmm::Element< TVR, TVM >, 461
 - gdcmm::Element< TVR, VM::VM1_2 >, 467

- gdcm::Element< TVR, VM::VM1_n >, [473](#)
- gdcm::Element< TVR, VM::VM2_2n >, [479](#)
- gdcm::Element< TVR, VM::VM2_n >, [485](#)
- gdcm::Element< TVR, VM::VM3_3n >, [491](#)
- gdcm::Element< TVR, VM::VM3_4 >, [497](#)
- gdcm::Element< TVR, VM::VM3_n >, [503](#)
- gdcm::Element< VR::AS, VM::VM5 >, [506](#)
- gdcm::Element< VR::OB, VM::VM1 >, [511](#)
- gdcm::Element< VR::OW, VM::VM1 >, [516](#)
- gdcm::EncodingImplementation< VR::VRASCII
>, [525](#), [526](#)
- gdcm::EncodingImplementation< VR::VRBINARY
>, [528](#)
- gdcm::ExplicitDataElement, [543](#)
- gdcm::File, [552](#)
- gdcm::FileAnonymizer, [557](#)
- gdcm::FileMetaInformation, [580](#)
- gdcm::Fragment, [614](#)
- gdcm::ImageWriter, [697](#)
- gdcm::ImplicitDataElement, [704](#)
- gdcm::Item, [725](#)
- gdcm::network::AAabortPDU, [115](#)
- gdcm::network::AAssociateACPDU, [119](#)
- gdcm::network::AAssociateRJPDU, [121](#)
- gdcm::network::AAssociateRQPDU, [127](#)
- gdcm::network::AbstractSyntax, [130](#)
- gdcm::network::ApplicationContext, [146](#)
- gdcm::network::AReleaseRPPDU, [151](#)
- gdcm::network::AReleaseRQPDU, [153](#)
- gdcm::network::AsynchronousOperationsWin-
dowSub, [158](#)
- gdcm::network::BasePDU, [231](#)
- gdcm::network::ImplementationClassUIDSub,
[698](#)
- gdcm::network::ImplementationUIDSub, [699](#)
- gdcm::network::ImplementationVersionName-
Sub, [700](#)
- gdcm::network::MaximumLengthSub, [792](#)
- gdcm::network::PDataTFPDU, [907](#)
- gdcm::network::PresentationContextAC, [972](#)
- gdcm::network::PresentationContextRQ, [978](#)
- gdcm::network::PresentationDataValue, [982](#)
- gdcm::network::RoleSelectionSub, [1048](#)
- gdcm::network::ServiceClassApplicationInfor-
mation, [1115](#)
- gdcm::network::SOPClassExtendedNegociation-
Sub, [1141](#)
- gdcm::network::TransferSyntaxSub, [1277](#)
- gdcm::network::UserInformation, [1410](#)
- gdcm::PGXCodec, [926](#)
- gdcm::PixmapWriter, [957](#)
- gdcm::PNMCodec, [962](#)
- gdcm::Preamble, [966](#)
- gdcm::SegmentWriter, [1091](#)
- gdcm::SequenceOfFragments, [1098](#)
- gdcm::SequenceOfItems, [1107](#)
- gdcm::StreamImageWriter, [1166](#)
- gdcm::SurfaceWriter, [1225](#)
- gdcm::Tag, [1254](#)
- gdcm::ValueIO< TDE, TSwap, TType >, [1417](#)
- gdcm::VL, [1424](#)
- gdcm::VR, [1436](#)
- gdcm::VRVLSIZE< 0 >, [1443](#)
- gdcm::VRVLSIZE< 1 >, [1444](#)
- gdcm::Writer, [1563](#)
- vtkGDCMImageWriter, [1481](#)
- Write16
 - gdcm::VL, [1424](#)
- WriteASCII
 - gdcm::Element< TVR, VM::VM1_n >, [473](#)
- WriteBuffer
 - gdcm::ByteValue, [283](#)
 - gdcm::SequenceOfFragments, [1099](#)
- WriteBufferAsRGBA
 - gdcm::LookupTable, [783](#)
- WriteData
 - vtkGDCMPolyDataWriter, [1493](#)
- WriteFooter
 - gdcm::DictConverter, [427](#)
- WriteGDCMData
 - vtkGDCMImageWriter, [1481](#)
- WriteHeader
 - gdcm::DictConverter, [427](#)
- WriteHelpFile
 - gdcm::BaseQuery, [235](#)
- WriteImageInformation
 - gdcm::StreamImageWriter, [1166](#)
- WriteImageSubregionRAW
 - gdcm::StreamImageWriter, [1167](#)
- WritePointer
 - vtkLookupTable16, [1542](#)
- WriteQuery
 - gdcm::BaseQuery, [235](#)
- Writer
 - gdcm::Writer, [1561](#)
- WriteRawHeader
 - gdcm::StreamImageWriter, [1167](#)
- WriteRTSTRUCTData
 - vtkGDCMPolyDataWriter, [1493](#)
- WriteRTSTRUCTInfo
 - vtkGDCMPolyDataWriter, [1494](#)
- WriteSlice
 - vtkGDCMImageWriter, [1482](#)
- x16printf
 - gdcm, [102](#)
- XAXRFGayscaleSoftcopyPresentationStateStorage
 - gdcm::UIDs, [1309](#)

XML
 gdcmm::Printer, 985
XMLDictReader
 gdcmm::XMLDictReader, 1566
XMLEncoding
 gdcmm::UIDs, 1302
XMLPrinter
 gdcmm::XMLPrinter, 1568
XMLPrivateDictReader
 gdcmm::XMLPrivateDictReader, 1572
XRay3DAngiographicImageStorage
 gdcmm::MediaStorage, 800
 gdcmm::UIDs, 1305
XRay3DCraniofacialImageStorage
 gdcmm::MediaStorage, 801
 gdcmm::UIDs, 1305
XRayAngiographicBiPlaneImageStorageRetired
 gdcmm::MediaStorage, 800
 gdcmm::UIDs, 1305
XRayAngiographicImageStorage
 gdcmm::MediaStorage, 800
 gdcmm::UIDs, 1305
XRayRadiationDoseSR
 gdcmm::MediaStorage, 801
XRayRadiationDoseSRStorage
 gdcmm::UIDs, 1306
XRayRadiofluoroscopicImageStorage
 gdcmm::UIDs, 1305
XRayRadiofluoroscopicImageStorage
 gdcmm::MediaStorage, 800

YBR2RGB
 gdcmm::ImageChangePhotometricInterpretation,
 644
YBR_FULL
 gdcmm::PhotometricInterpretation, 928
YBR_FULL_422
 gdcmm::PhotometricInterpretation, 928
YBR_ICT
 gdcmm::PhotometricInterpretation, 928
YBR_PARTIAL_420
 gdcmm::PhotometricInterpretation, 928
YBR_PARTIAL_422
 gdcmm::PhotometricInterpretation, 928
YBR_RCT
 gdcmm::PhotometricInterpretation, 928
yellow
 gdcmm::terminal, 110
YES
 gdcmm::Surface, 1204

ZEROED_OUT
 gdcmm::CSAHeader, 351
ZSpacing
 gdcmm::IPPSorter, 719

ZTolerance
 gdcmm::IPPSorter, 719