

GDCM

2.8.6

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>GDCM Documentation</b>	<b>1</b>
<b>2</b>	<b>Todo List</b>	<b>3</b>
<b>3</b>	<b>Deprecated List</b>	<b>5</b>
<b>4</b>	<b>Bug List</b>	<b>7</b>
<b>5</b>	<b>Namespace Index</b>	<b>9</b>
5.1	Namespace List . . . . .	9
<b>6</b>	<b>Hierarchical Index</b>	<b>11</b>
6.1	Class Hierarchy . . . . .	11
<b>7</b>	<b>Class Index</b>	<b>21</b>
7.1	Class List . . . . .	21
<b>8</b>	<b>File Index</b>	<b>35</b>
8.1	File List . . . . .	35

<b>9 Namespace Documentation</b>	<b>43</b>
9.1 gdcm Namespace Reference	43
9.1.1 Detailed Description	57
9.1.2 Typedef Documentation	58
9.1.2.1 AECComp	58
9.1.2.2 ASComp	58
9.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER	58
9.1.2.4 CSCComp	58
9.1.2.5 DComp	58
9.1.2.6 DTComp	58
9.1.2.7 FileList	59
9.1.2.8 IconImage	59
9.1.2.9 LOComp	59
9.1.2.10 LTComp	59
9.1.2.11 MacroEntry	59
9.1.2.12 NestedMacroEntries	59
9.1.2.13 PNComp	59
9.1.2.14 SHComp	60
9.1.2.15 STComp	60
9.1.2.16 TMComp	60
9.1.2.17 UIComp	60
9.1.2.18 UTComp	60
9.1.3 Enumeration Type Documentation	60
9.1.3.1 CompOperators	60
9.1.3.2 ECharSet	61
9.1.3.3 ENQueryType	61
9.1.3.4 EQueryLevel	61
9.1.3.5 EQueryType	62



9.1.3.6	ERootType	62
9.1.3.7	LodModeType	62
9.1.4	Function Documentation	63
9.1.4.1	backslash()	63
9.1.4.2	Clamp()	63
9.1.4.3	GetVRFromTag()	63
9.1.4.4	operator"!=() [1/2]	63
9.1.4.5	operator"!=() [2/2]	63
9.1.4.6	operator<<() [1/56]	64
9.1.4.7	operator<<() [2/56]	64
9.1.4.8	operator<<() [3/56]	64
9.1.4.9	operator<<() [4/56]	64
9.1.4.10	operator<<() [5/56]	64
9.1.4.11	operator<<() [6/56]	65
9.1.4.12	operator<<() [7/56]	65
9.1.4.13	operator<<() [8/56]	65
9.1.4.14	operator<<() [9/56]	65
9.1.4.15	operator<<() [10/56]	65
9.1.4.16	operator<<() [11/56]	66
9.1.4.17	operator<<() [12/56]	66
9.1.4.18	operator<<() [13/56]	66
9.1.4.19	operator<<() [14/56]	66
9.1.4.20	operator<<() [15/56]	66
9.1.4.21	operator<<() [16/56]	67
9.1.4.22	operator<<() [17/56]	67
9.1.4.23	operator<<() [18/56]	67
9.1.4.24	operator<<() [19/56]	67
9.1.4.25	operator<<() [20/56]	67

9.1.4.26	<code>operator&lt;&lt;()</code>	[21/56]	68
9.1.4.27	<code>operator&lt;&lt;()</code>	[22/56]	68
9.1.4.28	<code>operator&lt;&lt;()</code>	[23/56]	68
9.1.4.29	<code>operator&lt;&lt;()</code>	[24/56]	68
9.1.4.30	<code>operator&lt;&lt;()</code>	[25/56]	68
9.1.4.31	<code>operator&lt;&lt;()</code>	[26/56]	69
9.1.4.32	<code>operator&lt;&lt;()</code>	[27/56]	69
9.1.4.33	<code>operator&lt;&lt;()</code>	[28/56]	69
9.1.4.34	<code>operator&lt;&lt;()</code>	[29/56]	69
9.1.4.35	<code>operator&lt;&lt;()</code>	[30/56]	69
9.1.4.36	<code>operator&lt;&lt;()</code>	[31/56]	70
9.1.4.37	<code>operator&lt;&lt;()</code>	[32/56]	70
9.1.4.38	<code>operator&lt;&lt;()</code>	[33/56]	70
9.1.4.39	<code>operator&lt;&lt;()</code>	[34/56]	70
9.1.4.40	<code>operator&lt;&lt;()</code>	[35/56]	70
9.1.4.41	<code>operator&lt;&lt;()</code>	[36/56]	71
9.1.4.42	<code>operator&lt;&lt;()</code>	[37/56]	71
9.1.4.43	<code>operator&lt;&lt;()</code>	[38/56]	71
9.1.4.44	<code>operator&lt;&lt;()</code>	[39/56]	71
9.1.4.45	<code>operator&lt;&lt;()</code>	[40/56]	71
9.1.4.46	<code>operator&lt;&lt;()</code>	[41/56]	72
9.1.4.47	<code>operator&lt;&lt;()</code>	[42/56]	72
9.1.4.48	<code>operator&lt;&lt;()</code>	[43/56]	72
9.1.4.49	<code>operator&lt;&lt;()</code>	[44/56]	72
9.1.4.50	<code>operator&lt;&lt;()</code>	[45/56]	72
9.1.4.51	<code>operator&lt;&lt;()</code>	[46/56]	73
9.1.4.52	<code>operator&lt;&lt;()</code>	[47/56]	73
9.1.4.53	<code>operator&lt;&lt;()</code>	[48/56]	73

9.1.4.54	<a href="#">operator&lt;&lt;()</a> [49/56]	73
9.1.4.55	<a href="#">operator&lt;&lt;()</a> [50/56]	74
9.1.4.56	<a href="#">operator&lt;&lt;()</a> [51/56]	74
9.1.4.57	<a href="#">operator&lt;&lt;()</a> [52/56]	74
9.1.4.58	<a href="#">operator&lt;&lt;()</a> [53/56]	74
9.1.4.59	<a href="#">operator&lt;&lt;()</a> [54/56]	75
9.1.4.60	<a href="#">operator&lt;&lt;()</a> [55/56]	75
9.1.4.61	<a href="#">operator&lt;&lt;()</a> [56/56]	75
9.1.4.62	<a href="#">operator==()</a>	75
9.1.4.63	<a href="#">operator&gt;&gt;()</a> [1/3]	76
9.1.4.64	<a href="#">operator&gt;&gt;()</a> [2/3]	76
9.1.4.65	<a href="#">operator&gt;&gt;()</a> [3/3]	76
9.1.4.66	<a href="#">Round()</a>	76
9.1.4.67	<a href="#">to_string()</a>	76
9.1.4.68	<a href="#">TYPETOENCODING()</a>	77
9.1.5	<a href="#">Variable Documentation</a>	77
9.1.5.1	<a href="#">GlobalInstance</a>	77
9.1.5.2	<a href="#">VRBINARY</a>	77
9.2	<a href="#">gdcmm::network Namespace Reference</a>	77
9.2.1	<a href="#">Enumeration Type Documentation</a>	81
9.2.1.1	<a href="#">EEventID</a>	81
9.2.1.2	<a href="#">EStateID</a>	82
9.2.2	<a href="#">Function Documentation</a>	83
9.2.2.1	<a href="#">GetStateIndex()</a>	83
9.2.3	<a href="#">Variable Documentation</a>	83
9.2.3.1	<a href="#">cMaxEventID</a>	83
9.2.3.2	<a href="#">cMaxStateID</a>	83
9.3	<a href="#">gdcmm::SegmentHelper Namespace Reference</a>	84
9.4	<a href="#">gdcmm::terminal Namespace Reference</a>	84
9.4.1	<a href="#">Detailed Description</a>	84
9.4.2	<a href="#">Enumeration Type Documentation</a>	85
9.4.2.1	<a href="#">Attribute</a>	85
9.4.2.2	<a href="#">Color</a>	85
9.4.2.3	<a href="#">Mode</a>	85
9.4.3	<a href="#">Function Documentation</a>	86
9.4.3.1	<a href="#">setattribute()</a>	86
9.4.3.2	<a href="#">setbgcolor()</a>	86
9.4.3.3	<a href="#">setfgcolor()</a>	86
9.4.3.4	<a href="#">setmode()</a>	86

<b>10 Class Documentation</b>	<b>87</b>
10.1 <code>gdcn::network::AAAbortPDU</code> Class Reference	87
10.1.1 Detailed Description	88
10.1.2 Constructor & Destructor Documentation	88
10.1.2.1 <code>AAAbortPDU()</code>	88
10.1.3 Member Function Documentation	88
10.1.3.1 <code>IsLastFragment()</code>	88
10.1.3.2 <code>Print()</code>	88
10.1.3.3 <code>Read()</code>	89
10.1.3.4 <code>SetReason()</code>	89
10.1.3.5 <code>SetSource()</code>	89
10.1.3.6 <code>Size()</code>	89
10.1.3.7 <code>Write()</code>	89
10.2 <code>gdcn::network::AAssociateACPDU</code> Class Reference	90
10.2.1 Detailed Description	91
10.2.2 Member Typedef Documentation	91
10.2.2.1 <code>SizeType</code>	91
10.2.3 Constructor & Destructor Documentation	91
10.2.3.1 <code>AAssociateACPDU()</code>	92
10.2.4 Member Function Documentation	92
10.2.4.1 <code>AddPresentationContextAC()</code>	92
10.2.4.2 <code>GetNumberOfPresentationContextAC()</code>	92
10.2.4.3 <code>GetPresentationContextAC()</code>	92
10.2.4.4 <code>GetUserInformation()</code>	92
10.2.4.5 <code>InitFromRQ()</code>	92
10.2.4.6 <code>IsLastFragment()</code>	93
10.2.4.7 <code>Print()</code>	93
10.2.4.8 <code>Read()</code>	93

10.2.4.9 SetCalledAETitle()	93
10.2.4.10 SetCallingAETitle()	93
10.2.4.11 Size()	94
10.2.4.12 Write()	94
10.2.5 Friends And Related Function Documentation	94
10.2.5.1 AAssociateRQPDU	94
10.3 gdcm::network::AAssociateRJPDU Class Reference	94
10.3.1 Detailed Description	95
10.3.2 Constructor & Destructor Documentation	95
10.3.2.1 AAssociateRJPDU()	95
10.3.3 Member Function Documentation	95
10.3.3.1 IsLastFragment()	96
10.3.3.2 Print()	96
10.3.3.3 Read()	96
10.3.3.4 Size()	96
10.3.3.5 Write()	96
10.4 gdcm::network::AAssociateRQPDU Class Reference	97
10.4.1 Detailed Description	98
10.4.2 Member Typedef Documentation	98
10.4.2.1 PresentationContextArrayType	99
10.4.2.2 SizeType	99
10.4.3 Constructor & Destructor Documentation	99
10.4.3.1 AAssociateRQPDU() [1/2]	99
10.4.3.2 AAssociateRQPDU() [2/2]	99
10.4.4 Member Function Documentation	99
10.4.4.1 AddPresentationContext()	99
10.4.4.2 GetCalledAETitle()	99
10.4.4.3 GetCallingAETitle()	100

10.4.4.4	<a href="#">GetNumberOfPresentationContext()</a>	100
10.4.4.5	<a href="#">GetPresentationContext()</a>	100
10.4.4.6	<a href="#">GetPresentationContextByAbstractSyntax()</a>	100
10.4.4.7	<a href="#">GetPresentationContextByID()</a>	100
10.4.4.8	<a href="#">GetPresentationContexts()</a>	100
10.4.4.9	<a href="#">GetReserved43_74()</a>	101
10.4.4.10	<a href="#">GetUserInformation()</a>	101
10.4.4.11	<a href="#">IsAETitleValid()</a>	101
10.4.4.12	<a href="#">IsLastFragment()</a>	101
10.4.4.13	<a href="#">Print()</a>	101
10.4.4.14	<a href="#">Read()</a>	102
10.4.4.15	<a href="#">SetCalledAETitle()</a>	102
10.4.4.16	<a href="#">SetCallingAETitle()</a>	102
10.4.4.17	<a href="#">SetUserInformation()</a>	102
10.4.4.18	<a href="#">Size()</a>	102
10.4.4.19	<a href="#">Write()</a>	103
10.4.5	<a href="#">Friends And Related Function Documentation</a>	103
10.4.5.1	<a href="#">AAssociateACPDU</a>	103
10.5	<a href="#">gdcm::AbortEvent Class Reference</a>	103
10.6	<a href="#">gdcm::network::AbstractSyntax Class Reference</a>	104
10.6.1	<a href="#">Detailed Description</a>	105
10.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	105
10.6.2.1	<a href="#">AbstractSyntax()</a>	105
10.6.3	<a href="#">Member Function Documentation</a>	105
10.6.3.1	<a href="#">GetAsDataElement()</a>	105
10.6.3.2	<a href="#">GetName()</a>	105
10.6.3.3	<a href="#">operator==(())</a>	105
10.6.3.4	<a href="#">Print()</a>	106

10.6.3.5	Read()	106
10.6.3.6	SetName()	106
10.6.3.7	SetNameFromUID()	106
10.6.3.8	Size()	106
10.6.3.9	Write()	106
10.7	gdcm::AnonymizeEvent Class Reference	107
10.7.1	Detailed Description	108
10.7.2	Member Typedef Documentation	108
10.7.2.1	Self	108
10.7.2.2	Superclass	108
10.7.3	Constructor & Destructor Documentation	108
10.7.3.1	AnonymizeEvent() [1/2]	108
10.7.3.2	~AnonymizeEvent()	109
10.7.3.3	AnonymizeEvent() [2/2]	109
10.7.4	Member Function Documentation	109
10.7.4.1	CheckEvent()	109
10.7.4.2	GetEventName()	109
10.7.4.3	GetTag()	109
10.7.4.4	MakeObject()	109
10.7.4.5	SetTag()	110
10.8	gdcm::Anonymizer Class Reference	110
10.8.1	Detailed Description	112
10.8.2	Constructor & Destructor Documentation	113
10.8.2.1	Anonymizer()	113
10.8.2.2	~Anonymizer()	113
10.8.3	Member Function Documentation	113
10.8.3.1	BALCPPProtect()	113
10.8.3.2	BasicApplicationLevelConfidentialityProfile()	114

10.8.3.3 CanEmptyTag()	114
10.8.3.4 ClearInternalUIDs()	114
10.8.3.5 Empty()	114
10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes()	115
10.8.3.7 GetCryptographicMessageSyntax()	115
10.8.3.8 GetFile()	115
10.8.3.9 New()	115
10.8.3.10 RecurseDataSet()	115
10.8.3.11 Remove()	115
10.8.3.12 RemoveGroupLength()	116
10.8.3.13 RemovePrivateTags()	116
10.8.3.14 RemoveRetired()	116
10.8.3.15 Replace() [1/2]	116
10.8.3.16 Replace() [2/2]	117
10.8.3.17 SetCryptographicMessageSyntax()	117
10.8.3.18 SetFile()	117
10.9 gdcm::AnyEvent Class Reference	118
10.10gdcm::network::ApplicationContext Class Reference	119
10.10.1 Detailed Description	119
10.10.2 Constructor & Destructor Documentation	120
10.10.2.1 ApplicationContext()	120
10.10.3 Member Function Documentation	120
10.10.3.1 GetName()	120
10.10.3.2 Print()	120
10.10.3.3 Read()	120
10.10.3.4 SetName()	120
10.10.3.5 Size()	121
10.10.3.6 Write()	121



10.11gdcmm::ApplicationEntity Class Reference . . . . .	121
10.11.1 Detailed Description . . . . .	122
10.11.2 Member Function Documentation . . . . .	122
10.11.2.1 IsValid() . . . . .	122
10.11.2.2 Print() . . . . .	123
10.11.2.3 SetBlob() . . . . .	123
10.11.2.4 Squeeze() . . . . .	123
10.11.3 Member Data Documentation . . . . .	123
10.11.3.1 Internal . . . . .	123
10.11.3.2 MaxLength . . . . .	123
10.11.3.3 MaxNumberOfComponents . . . . .	123
10.11.3.4 Padding . . . . .	124
10.11.3.5 Separator . . . . .	124
10.12gdcmm::network::AReleaseRPPDU Class Reference . . . . .	124
10.12.1 Detailed Description . . . . .	125
10.12.2 Constructor & Destructor Documentation . . . . .	125
10.12.2.1 AReleaseRPPDU() . . . . .	125
10.12.3 Member Function Documentation . . . . .	125
10.12.3.1 IsLastFragment() . . . . .	125
10.12.3.2 Print() . . . . .	125
10.12.3.3 Read() . . . . .	126
10.12.3.4 Size() . . . . .	126
10.12.3.5 Write() . . . . .	126
10.13gdcmm::network::AReleaseRQPDU Class Reference . . . . .	126
10.13.1 Detailed Description . . . . .	127
10.13.2 Constructor & Destructor Documentation . . . . .	127
10.13.2.1 AReleaseRQPDU() . . . . .	127
10.13.3 Member Function Documentation . . . . .	127

10.13.3.1 IsLastFragment()	128
10.13.3.2 Print()	128
10.13.3.3 Read()	128
10.13.3.4 Size()	128
10.13.3.5 Write()	128
10.14gdcm::network::ARTIMTimer Class Reference	129
10.14.1 Detailed Description	129
10.14.2 Constructor & Destructor Documentation	129
10.14.2.1 ARTIMTimer()	129
10.14.3 Member Function Documentation	129
10.14.3.1 GetElapsedTime()	130
10.14.3.2 GetHasExpired()	130
10.14.3.3 GetTimeout()	130
10.14.3.4 SetTimeout()	130
10.14.3.5 Start()	130
10.14.3.6 Stop()	130
10.15gdcm::ASN1 Class Reference	130
10.15.1 Detailed Description	131
10.15.2 Constructor & Destructor Documentation	131
10.15.2.1 ASN1()	131
10.15.2.2 ~ASN1()	131
10.15.3 Member Function Documentation	131
10.15.3.1 ParseDump()	131
10.15.3.2 ParseDumpFile()	132
10.15.3.3 TestPBKDF2()	132
10.16gdcm::network::AsynchronousOperationsWindowSub Class Reference	132
10.16.1 Detailed Description	132
10.16.2 Constructor & Destructor Documentation	132

10.16.2.1 AsynchronousOperationsWindowSub()	132
10.16.3 Member Function Documentation	133
10.16.3.1 Print()	133
10.16.3.2 Read()	133
10.16.3.3 Size()	133
10.16.3.4 Write()	133
10.17gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference	134
10.17.1 Detailed Description	135
10.17.2 Member Typedef Documentation	136
10.17.2.1 ArrayType	136
10.17.3 Member Enumeration Documentation	136
10.17.3.1 anonymous enum	136
10.17.4 Member Function Documentation	136
10.17.4.1 GDCM_STATIC_ASSERT() [1/3]	136
10.17.4.2 GDCM_STATIC_ASSERT() [2/3]	136
10.17.4.3 GDCM_STATIC_ASSERT() [3/3]	137
10.17.4.4 GetAsDataElement()	137
10.17.4.5 GetDictVM()	137
10.17.4.6 GetDictVR()	137
10.17.4.7 GetNumberOfValues()	137
10.17.4.8 GetTag()	138
10.17.4.9 GetValue() [1/2]	138
10.17.4.10GetValue() [2/2]	138
10.17.4.11GetValues()	138
10.17.4.12GetVM()	138
10.17.4.13GetVR()	139
10.17.4.14operator"!="()	139
10.17.4.15operator<()	139

10.17.4.16operator==( )	139
10.17.4.17operator[]( ) [ 1 / 2 ]	139
10.17.4.18operator[]( ) [ 2 / 2 ]	140
10.17.4.19Print( )	140
10.17.4.20Set( )	140
10.17.4.21SetByteValue( )	140
10.17.4.22SetByteValueNoSwap( )	140
10.17.4.23SetFromDataElement( )	141
10.17.4.24SetFromDataSet( )	141
10.17.4.25SetValue( )	141
10.17.4.26SetValues( )	141
10.17.5 Member Data Documentation	142
10.17.5.1 Internal	142
10.18gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	142
10.18.1 Member Typedef Documentation	143
10.18.1.1 ArrayType	143
10.18.2 Member Enumeration Documentation	144
10.18.2.1 anonymous enum	144
10.18.3 Member Function Documentation	144
10.18.3.1 GDCM_STATIC_ASSERT( ) [ 1 / 4 ]	144
10.18.3.2 GDCM_STATIC_ASSERT( ) [ 2 / 4 ]	144
10.18.3.3 GDCM_STATIC_ASSERT( ) [ 3 / 4 ]	144
10.18.3.4 GDCM_STATIC_ASSERT( ) [ 4 / 4 ]	145
10.18.3.5 GetAsDataElement( )	145
10.18.3.6 GetDictVM( )	145
10.18.3.7 GetDictVR( )	145
10.18.3.8 GetNumberOfValues( )	145
10.18.3.9 GetTag( )	145

10.18.3.10	<a href="#">GetValue() [1/2]</a>	146
10.18.3.11	<a href="#">GetValue() [2/2]</a>	146
10.18.3.12	<a href="#">GetValues()</a>	146
10.18.3.13	<a href="#">GetVM()</a>	146
10.18.3.14	<a href="#">GetVR()</a>	146
10.18.3.15	<a href="#">operator"!="()</a>	146
10.18.3.16	<a href="#">operator&lt;()</a>	147
10.18.3.17	<a href="#">operator==()</a>	147
10.18.3.18	<a href="#">Print()</a>	147
10.18.3.19	<a href="#">Set()</a>	147
10.18.3.20	<a href="#">SetByteValue()</a>	147
10.18.3.21	<a href="#">SetByteValueNoSwap()</a>	148
10.18.3.22	<a href="#">SetFromDataElement()</a>	148
10.18.3.23	<a href="#">SetFromDataSet()</a>	148
10.18.3.24	<a href="#">SetValue()</a>	148
10.18.4	<a href="#">Member Data Documentation</a>	148
10.18.4.1	<a href="#">Internal</a>	149
10.19	<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_3 &gt; Class Template Reference</a>	149
10.19.1	<a href="#">Member Function Documentation</a>	150
10.19.1.1	<a href="#">GetVM()</a>	150
10.20	<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_8 &gt; Class Template Reference</a>	150
10.20.1	<a href="#">Member Function Documentation</a>	151
10.20.1.1	<a href="#">GetVM()</a>	151
10.21	<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_n &gt; Class Template Reference</a>	152
10.21.1	<a href="#">Member Typedef Documentation</a>	153
10.21.1.1	<a href="#">ArrayType</a>	153
10.21.2	<a href="#">Constructor &amp; Destructor Documentation</a>	153
10.21.2.1	<a href="#">Attribute()</a>	153

10.21.2.2 ~Attribute()	153
10.21.3 Member Function Documentation	153
10.21.3.1 GDCM_STATIC_ASSERT() [1/3]	154
10.21.3.2 GDCM_STATIC_ASSERT() [2/3]	154
10.21.3.3 GDCM_STATIC_ASSERT() [3/3]	154
10.21.3.4 GetAsDataElement()	154
10.21.3.5 GetDictVM()	154
10.21.3.6 GetDictVR()	154
10.21.3.7 GetNumberOfValues()	155
10.21.3.8 GetTag()	155
10.21.3.9 GetValue() [1/2]	155
10.21.3.10 GetValue() [2/2]	155
10.21.3.11 GetValues()	155
10.21.3.12 GetVM()	155
10.21.3.13 GetVR()	156
10.21.3.14 operator[]() [1/2]	156
10.21.3.15 operator[]() [2/2]	156
10.21.3.16 Print()	156
10.21.3.17 Set()	156
10.21.3.18 SetByteValue()	157
10.21.3.19 SetFromDataElement()	157
10.21.3.20 SetFromDataSet()	157
10.21.3.21 SetNumberOfValues()	157
10.21.3.22 SetValue() [1/2]	157
10.21.3.23 SetValue() [2/2]	158
10.21.3.24 SetValues()	158
10.22 gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference	158
10.22.1 Member Function Documentation	159

10.22.1.1 GetVM()	159
10.23gdcM::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	160
10.23.1 Member Function Documentation	161
10.23.1.1 GetVM()	161
10.24gdcM::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	161
10.24.1 Member Function Documentation	162
10.24.1.1 GetVM()	162
10.25gdcM::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	163
10.25.1 Member Function Documentation	164
10.25.1.1 GetVM()	164
10.26gdcM::AudioCodec Class Reference	164
10.26.1 Detailed Description	165
10.26.2 Constructor & Destructor Documentation	165
10.26.2.1 AudioCodec()	165
10.26.2.2 ~AudioCodec()	166
10.26.3 Member Function Documentation	166
10.26.3.1 CanCode()	166
10.26.3.2 CanDecode()	166
10.26.3.3 Decode()	166
10.27gdcM::Base64 Class Reference	167
10.27.1 Detailed Description	167
10.27.2 Member Function Documentation	167
10.27.2.1 Decode()	167
10.27.2.2 Encode()	168
10.27.2.3 GetDecodeLength()	168
10.27.2.4 GetEncodeLength()	169
10.28gdcM::network::BaseCompositeMessage Class Reference	169
10.28.1 Detailed Description	170

10.28.2 Constructor & Destructor Documentation . . . . .	170
10.28.2.1 ~BaseCompositeMessage() . . . . .	170
10.28.3 Member Function Documentation . . . . .	170
10.28.3.1 ConstructPDV() . . . . .	171
10.29gdcm::network::BaseNormalizedMessage Class Reference . . . . .	171
10.29.1 Detailed Description . . . . .	172
10.29.2 Constructor & Destructor Documentation . . . . .	172
10.29.2.1 ~BaseNormalizedMessage() . . . . .	172
10.29.3 Member Function Documentation . . . . .	172
10.29.3.1 ConstructPDV() . . . . .	173
10.30gdcm::network::BasePDU Class Reference . . . . .	173
10.30.1 Detailed Description . . . . .	174
10.30.2 Constructor & Destructor Documentation . . . . .	174
10.30.2.1 ~BasePDU() . . . . .	174
10.30.3 Member Function Documentation . . . . .	174
10.30.3.1 IsLastFragment() . . . . .	175
10.30.3.2 Print() . . . . .	175
10.30.3.3 Read() . . . . .	175
10.30.3.4 Size() . . . . .	175
10.30.3.5 Write() . . . . .	176
10.31gdcm::BaseQuery Class Reference . . . . .	176
10.31.1 Detailed Description . . . . .	178
10.31.2 Constructor & Destructor Documentation . . . . .	178
10.31.2.1 BaseQuery() . . . . .	178
10.31.2.2 ~BaseQuery() . . . . .	178
10.31.3 Member Function Documentation . . . . .	178
10.31.3.1 AddQueryDataSet() . . . . .	178
10.31.3.2 GetAbstractSyntaxUID() . . . . .	178



10.31.3.3 GetQueryDataSet() [1/2]	179
10.31.3.4 GetQueryDataSet() [2/2]	179
10.31.3.5 GetSOPInstanceUID()	179
10.31.3.6 Print()	179
10.31.3.7 SetSearchParameter() [1/3]	179
10.31.3.8 SetSearchParameter() [2/3]	179
10.31.3.9 SetSearchParameter() [3/3]	180
10.31.3.10SetSOPInstanceUID()	180
10.31.3.11ValidateQuery()	180
10.31.3.12ValidDataSet()	180
10.31.3.13WriteHelpFile()	180
10.31.3.14WriteQuery()	180
10.31.4 Friends And Related Function Documentation	181
10.31.4.1 QueryFactory	181
10.31.5 Member Data Documentation	181
10.31.5.1 mDataSet	181
10.31.5.2 mHelpDescription	181
10.31.5.3 mSopInstanceUID	181
10.32gdcmm::BaseRootQuery Class Reference	182
10.32.1 Detailed Description	183
10.32.2 Constructor & Destructor Documentation	183
10.32.2.1 BaseRootQuery()	183
10.32.2.2 ~BaseRootQuery()	183
10.32.3 Member Function Documentation	184
10.32.3.1 Construct()	184
10.32.3.2 GetQueryLevelFromQueryRoot()	184
10.32.3.3 GetQueryLevelFromString()	184
10.32.3.4 GetQueryLevelString()	184

10.32.3.5 GetTagListByLevel()	184
10.32.3.6 InitializeDataSet()	185
10.32.3.7 ValidateQuery()	185
10.32.4 Friends And Related Function Documentation	185
10.32.4.1 QueryFactory	185
10.32.5 Member Data Documentation	185
10.32.5.1 mHelpDescription	186
10.32.5.2 mImage	186
10.32.5.3 mPatient	186
10.32.5.4 mRootType	186
10.32.5.5 mSeries	186
10.32.5.6 mStudy	186
10.33gdcmm::SegmentHelper::BasicCodedEntry Struct Reference	187
10.33.1 Detailed Description	188
10.33.2 Constructor & Destructor Documentation	188
10.33.2.1 BasicCodedEntry() [1/3]	188
10.33.2.2 BasicCodedEntry() [2/3]	188
10.33.2.3 BasicCodedEntry() [3/3]	188
10.33.3 Member Function Documentation	188
10.33.3.1 IsEmpty()	188
10.33.4 Member Data Documentation	189
10.33.4.1 CM	189
10.33.4.2 CSD	189
10.33.4.3 CSV	189
10.33.4.4 CV	189
10.34gdcmm::BasicOffsetTable Class Reference	190
10.34.1 Detailed Description	191
10.34.2 Constructor & Destructor Documentation	191

10.34.2.1 BasicOffsetTable()	191
10.34.3 Member Function Documentation	191
10.34.3.1 Read()	191
10.34.4 Friends And Related Function Documentation	191
10.34.4.1 operator<<	192
10.35gdcm::Bitmap Class Reference	192
10.35.1 Detailed Description	195
10.35.2 Member Typedef Documentation	195
10.35.2.1 LUTPtr	195
10.35.3 Constructor & Destructor Documentation	195
10.35.3.1 Bitmap()	195
10.35.3.2 ~Bitmap()	195
10.35.4 Member Function Documentation	195
10.35.4.1 AreOverlaysInPixelData()	196
10.35.4.2 Clear()	196
10.35.4.3 ComputeLossyFlag()	196
10.35.4.4 GetBuffer()	196
10.35.4.5 GetBuffer2()	196
10.35.4.6 GetBufferLength()	197
10.35.4.7 GetColumns()	197
10.35.4.8 GetDataElement() [1/2]	197
10.35.4.9 GetDataElement() [2/2]	197
10.35.4.10GetDimension()	197
10.35.4.11GetDimensions()	198
10.35.4.12GetLUT() [1/2]	198
10.35.4.13GetLUT() [2/2]	198
10.35.4.14GetNeedByteSwap()	198
10.35.4.15GetNumberOfDimensions()	198

10.35.4.16GetPhotometricInterpretation()	199
10.35.4.17GetPixelFormat() [1/2]	199
10.35.4.18GetPixelFormat() [2/2]	199
10.35.4.19GetPlanarConfiguration()	199
10.35.4.20GetRows()	199
10.35.4.21GetTransferSyntax()	200
10.35.4.22IsEmpty()	200
10.35.4.23IsLossy()	200
10.35.4.24IsTransferSyntaxCompatible()	200
10.35.4.25Print()	200
10.35.4.26SetColumns()	201
10.35.4.27SetDataElement()	201
10.35.4.28SetDimension()	201
10.35.4.29SetDimensions()	201
10.35.4.30SetLossyFlag()	202
10.35.4.31SetLUT()	202
10.35.4.32SetNeedByteSwap()	202
10.35.4.33SetNumberOfDimensions()	202
10.35.4.34SetPhotometricInterpretation()	202
10.35.4.35SetPixelFormat()	203
10.35.4.36SetPlanarConfiguration()	203
10.35.4.37SetRows()	203
10.35.4.38SetTransferSyntax()	203
10.35.4.39TryJPEG2000Codec()	204
10.35.4.40TryJPEG2000Codec2()	204
10.35.4.41TryJPEGCodec()	204
10.35.4.42TryJPEGCodec2()	204
10.35.4.43TryJPEGLSCodec()	204

10.35.4.44TryKAKADUCodec()	204
10.35.4.45TryPVRGCodec()	205
10.35.4.46TryRAWCodec()	205
10.35.4.47TryRLECodec()	205
10.35.4.48UnusedBitsPresentInPixelData()	205
10.35.5 Friends And Related Function Documentation	205
10.35.5.1 ImageChangeTransferSyntax	205
10.35.5.2 PixmapReader	206
10.35.6 Member Data Documentation	206
10.35.6.1 Dimensions	206
10.35.6.2 LossyFlag	206
10.35.6.3 LUT	206
10.35.6.4 NeedByteSwap	206
10.35.6.5 NumberOfDimensions	206
10.35.6.6 PF	206
10.35.6.7 PI	207
10.35.6.8 PixelData	207
10.35.6.9 PlanarConfiguration	207
10.35.6.10TS	207
10.36gdcm::BitmapToBitmapFilter Class Reference	207
10.36.1 Detailed Description	208
10.36.2 Constructor & Destructor Documentation	209
10.36.2.1 BitmapToBitmapFilter()	209
10.36.2.2 ~BitmapToBitmapFilter()	209
10.36.3 Member Function Documentation	209
10.36.3.1 GetOutput()	209
10.36.3.2 GetOutputAsBitmap()	209
10.36.3.3 SetInput()	209

10.36.4 Member Data Documentation . . . . .	210
10.36.4.1 Input . . . . .	210
10.36.4.2 Output . . . . .	210
10.37gdcm::BoxRegion Class Reference . . . . .	210
10.37.1 Detailed Description . . . . .	212
10.37.2 Constructor & Destructor Documentation . . . . .	212
10.37.2.1 BoxRegion() [1/2] . . . . .	212
10.37.2.2 ~BoxRegion() . . . . .	212
10.37.2.3 BoxRegion() [2/2] . . . . .	212
10.37.3 Member Function Documentation . . . . .	212
10.37.3.1 Area() . . . . .	212
10.37.3.2 BoundingBox() . . . . .	213
10.37.3.3 Clone() . . . . .	213
10.37.3.4 ComputeBoundingBox() . . . . .	213
10.37.3.5 Empty() . . . . .	213
10.37.3.6 GetXMax() . . . . .	213
10.37.3.7 GetXMin() . . . . .	214
10.37.3.8 GetYMax() . . . . .	214
10.37.3.9 GetYMin() . . . . .	214
10.37.3.10GetZMax() . . . . .	214
10.37.3.11GetZMin() . . . . .	214
10.37.3.12IsValid() . . . . .	214
10.37.3.13operator=() . . . . .	215
10.37.3.14Print() . . . . .	215
10.37.3.15SetDomain() . . . . .	215
10.38gdcm::ByteBuffer Class Reference . . . . .	215
10.38.1 Detailed Description . . . . .	216
10.38.2 Constructor & Destructor Documentation . . . . .	216

10.38.2.1 ByteBuffer()	216
10.38.3 Member Function Documentation	216
10.38.3.1 Get()	216
10.38.3.2 GetStart()	216
10.38.3.3 ShiftEnd()	216
10.38.3.4 UpdatePosition()	217
10.39gdcm::ByteSwap< T > Class Template Reference	217
10.39.1 Detailed Description	217
10.39.2 Member Function Documentation	217
10.39.2.1 Swap()	218
10.39.2.2 SwapFromSwapCodeIntoSystem()	218
10.39.2.3 SwapRange()	218
10.39.2.4 SwapRangeFromSwapCodeIntoSystem()	218
10.39.2.5 SystemIsBigEndian()	219
10.39.2.6 SystemIsLittleEndian()	219
10.40gdcm::ByteSwapFilter Class Reference	219
10.40.1 Detailed Description	219
10.40.2 Constructor & Destructor Documentation	219
10.40.2.1 ByteSwapFilter()	220
10.40.2.2 ~ByteSwapFilter()	220
10.40.3 Member Function Documentation	220
10.40.3.1 ByteSwap()	220
10.40.3.2 SetByteSwapTag()	220
10.41gdcm::ByteValue Class Reference	221
10.41.1 Detailed Description	222
10.41.2 Constructor & Destructor Documentation	223
10.41.2.1 ByteValue() [1/2]	223
10.41.2.2 ByteValue() [2/2]	223

10.41.2.3 ~ByteValue()	223
10.41.3 Member Function Documentation	223
10.41.3.1 Append()	224
10.41.3.2 Clear()	224
10.41.3.3 ComputeLength()	224
10.41.3.4 Fill()	224
10.41.3.5 GetBuffer()	224
10.41.3.6 GetLength()	225
10.41.3.7 GetPointer()	225
10.41.3.8 IsEmpty()	225
10.41.3.9 IsPrintable()	226
10.41.3.10 operator const std::vector< char > &()	226
10.41.3.11 operator=()	226
10.41.3.12 operator==( ) [1/2]	226
10.41.3.13 operator==( ) [2/2]	226
10.41.3.14 Print()	226
10.41.3.15 PrintASCII()	227
10.41.3.16 PrintASCIIXML()	227
10.41.3.17 PrintGroupLength()	227
10.41.3.18 PrintHex()	227
10.41.3.19 PrintHexXML()	227
10.41.3.20 PrintPNXML()	227
10.41.3.21 Read() [1/2]	228
10.41.3.22 Read() [2/2]	228
10.41.3.23 SetLength()	228
10.41.3.24 SetLengthOnly()	228
10.41.3.25 Write() [1/2]	228
10.41.3.26 Write() [2/2]	229



10.41.3.27WriteBuffer()	229
10.42gdcmm::CAPICryptoFactory Class Reference	229
10.42.1 Constructor & Destructor Documentation	230
10.42.1.1 CAPICryptoFactory()	230
10.42.2 Member Function Documentation	230
10.42.2.1 CreateCMSProvider()	230
10.43gdcmm::CAPICryptographicMessageSyntax Class Reference	231
10.43.1 Constructor & Destructor Documentation	232
10.43.1.1 CAPICryptographicMessageSyntax()	232
10.43.1.2 ~CAPICryptographicMessageSyntax()	232
10.43.2 Member Function Documentation	232
10.43.2.1 Decrypt()	232
10.43.2.2 Encrypt()	233
10.43.2.3 GetCipherType()	233
10.43.2.4 GetInitialized()	233
10.43.2.5 ParseCertificateFile()	233
10.43.2.6 ParseKeyFile()	233
10.43.2.7 SetCipherType()	234
10.43.2.8 SetPassword()	234
10.44gdcmm::network::CEchoRQ Class Reference	234
10.44.1 Detailed Description	235
10.44.2 Member Function Documentation	235
10.44.2.1 ConstructPDV()	235
10.44.3 Member Data Documentation	236
10.44.3.1 AffectedSOPClassUID	236
10.44.3.2 MessageID	236
10.45gdcmm::network::CEchoRSP Class Reference	236
10.45.1 Detailed Description	237

10.45.2 Member Function Documentation . . . . .	237
10.45.2.1 ConstructPDVByDataSet() . . . . .	237
10.46gdcm::network::CFind Class Reference . . . . .	237
10.46.1 Detailed Description . . . . .	238
10.47gdcm::network::CFindCancelRQ Class Reference . . . . .	238
10.47.1 Detailed Description . . . . .	239
10.47.2 Member Function Documentation . . . . .	239
10.47.2.1 ConstructPDVByDataSet() . . . . .	239
10.48gdcm::network::CFindRQ Class Reference . . . . .	239
10.48.1 Detailed Description . . . . .	240
10.48.2 Member Function Documentation . . . . .	240
10.48.2.1 ConstructPDV() . . . . .	240
10.49gdcm::network::CFindRSP Class Reference . . . . .	241
10.49.1 Detailed Description . . . . .	241
10.49.2 Member Function Documentation . . . . .	242
10.49.2.1 ConstructPDVByDataSet() . . . . .	242
10.50gdcm::network::CMoveCancelRq Class Reference . . . . .	242
10.50.1 Member Function Documentation . . . . .	243
10.50.1.1 ConstructPDVByDataSet() . . . . .	243
10.51gdcm::network::CMoveRQ Class Reference . . . . .	243
10.51.1 Detailed Description . . . . .	244
10.51.2 Member Function Documentation . . . . .	244
10.51.2.1 ConstructPDV() . . . . .	244
10.52gdcm::network::CMoveRSP Class Reference . . . . .	245
10.52.1 Detailed Description . . . . .	245
10.52.2 Member Function Documentation . . . . .	246
10.52.2.1 ConstructPDVByDataSet() . . . . .	246
10.53gdcm::Codec Class Reference . . . . .	246

10.53.1 Detailed Description . . . . .	247
10.54gdcm::Coder Class Reference . . . . .	247
10.54.1 Detailed Description . . . . .	248
10.54.2 Constructor & Destructor Documentation . . . . .	248
10.54.2.1 ~Coder() . . . . .	248
10.54.3 Member Function Documentation . . . . .	248
10.54.3.1 CanCode() . . . . .	248
10.54.3.2 Code() . . . . .	249
10.54.3.3 InternalCode() . . . . .	249
10.55gdcm::CodeString Class Reference . . . . .	249
10.55.1 Detailed Description . . . . .	250
10.55.2 Member Typedef Documentation . . . . .	250
10.55.2.1 const_iterator . . . . .	251
10.55.2.2 const_reference . . . . .	251
10.55.2.3 const_reverse_iterator . . . . .	251
10.55.2.4 difference_type . . . . .	251
10.55.2.5 iterator . . . . .	251
10.55.2.6 pointer . . . . .	251
10.55.2.7 reference . . . . .	251
10.55.2.8 reverse_iterator . . . . .	252
10.55.2.9 size_type . . . . .	252
10.55.2.10value_type . . . . .	252
10.55.3 Constructor & Destructor Documentation . . . . .	252
10.55.3.1 CodeString() [1/4] . . . . .	252
10.55.3.2 CodeString() [2/4] . . . . .	252
10.55.3.3 CodeString() [3/4] . . . . .	252
10.55.3.4 CodeString() [4/4] . . . . .	253
10.55.4 Member Function Documentation . . . . .	253

10.55.4.1 GetAsString()	253
10.55.4.2 IsValid()	253
10.55.4.3 Size()	253
10.55.4.4 TrimInternal()	253
10.55.5 Friends And Related Function Documentation	253
10.55.5.1 operator"!="	254
10.55.5.2 operator<<	254
10.55.5.3 operator==	254
10.56gdcmm::Command Class Reference	254
10.56.1 Detailed Description	255
10.56.2 Constructor & Destructor Documentation	255
10.56.2.1 Command()	256
10.56.2.2 ~Command()	256
10.56.3 Member Function Documentation	256
10.56.3.1 Execute() [1/2]	256
10.56.3.2 Execute() [2/2]	256
10.57gdcmm::CommandDataSet Class Reference	257
10.57.1 Detailed Description	258
10.57.2 Constructor & Destructor Documentation	258
10.57.2.1 CommandDataSet()	258
10.57.2.2 ~CommandDataSet()	258
10.57.3 Member Function Documentation	258
10.57.3.1 Insert()	258
10.57.3.2 Read()	259
10.57.3.3 Replace()	259
10.57.3.4 Write()	259
10.57.4 Friends And Related Function Documentation	259
10.57.4.1 operator<<	259

10.58gdcmm::network::CompositeMessageFactory Class Reference	259
10.58.1 Detailed Description	260
10.58.2 Member Function Documentation	260
10.58.2.1 ConstructCEchoRQ()	260
10.58.2.2 ConstructCFindRQ()	260
10.58.2.3 ConstructCMoveRQ()	261
10.58.2.4 ConstructCStoreRQ()	261
10.58.2.5 ConstructCStoreRSP()	261
10.59gdcmm::CompositeNetworkFunctions Class Reference	261
10.59.1 Detailed Description	262
10.59.2 Member Typedef Documentation	262
10.59.2.1 KeyValuePairArrayType	262
10.59.2.2 KeyValuePairType	262
10.59.3 Member Function Documentation	263
10.59.3.1 CEcho()	263
10.59.3.2 CFind()	263
10.59.3.3 CMove()	264
10.59.3.4 ConstructQuery() [1/2]	264
10.59.3.5 ConstructQuery() [2/2]	265
10.59.3.6 CStore()	265
10.60gdcmm::ConstCharWrapper Class Reference	266
10.60.1 Detailed Description	266
10.60.2 Constructor & Destructor Documentation	266
10.60.2.1 ConstCharWrapper()	266
10.60.3 Member Function Documentation	266
10.60.3.1 operator const char *()	266
10.61gdcmm::CP246ExplicitDataElement Class Reference	267
10.61.1 Detailed Description	268

10.61.2 Member Function Documentation . . . . .	268
10.61.2.1 GetLength() . . . . .	268
10.61.2.2 Read() . . . . .	268
10.61.2.3 ReadPreValue() . . . . .	268
10.61.2.4 ReadValue() . . . . .	269
10.61.2.5 ReadWithLength() . . . . .	269
10.62gdcmm::CryptoFactory Class Reference . . . . .	269
10.62.1 Detailed Description . . . . .	270
10.62.2 Member Enumeration Documentation . . . . .	270
10.62.2.1 CryptoLib . . . . .	270
10.62.3 Constructor & Destructor Documentation . . . . .	271
10.62.3.1 CryptoFactory() [1/2] . . . . .	271
10.62.3.2 CryptoFactory() [2/2] . . . . .	271
10.62.3.3 ~CryptoFactory() . . . . .	271
10.62.4 Member Function Documentation . . . . .	271
10.62.4.1 CreateCMSProvider() . . . . .	271
10.62.4.2 GetFactoryInstance() . . . . .	271
10.63gdcmm::CryptographicMessageSyntax Class Reference . . . . .	272
10.63.1 Member Enumeration Documentation . . . . .	272
10.63.1.1 CipherTypes . . . . .	272
10.63.2 Constructor & Destructor Documentation . . . . .	273
10.63.2.1 CryptographicMessageSyntax() . . . . .	273
10.63.2.2 ~CryptographicMessageSyntax() . . . . .	273
10.63.3 Member Function Documentation . . . . .	273
10.63.3.1 Decrypt() . . . . .	273
10.63.3.2 Encrypt() . . . . .	274
10.63.3.3 GetCipherType() . . . . .	274
10.63.3.4 ParseCertificateFile() . . . . .	274

10.63.3.5 ParseKeyFile()	274
10.63.3.6 SetCipherType()	275
10.63.3.7 SetPassword()	275
10.64gdcM::CSAElement Class Reference	275
10.64.1 Detailed Description	277
10.64.2 Member Typedef Documentation	277
10.64.2.1 DataPtr	277
10.64.3 Constructor & Destructor Documentation	277
10.64.3.1 CSAElement() [1/2]	277
10.64.3.2 CSAElement() [2/2]	277
10.64.4 Member Function Documentation	278
10.64.4.1 GetByteValue()	278
10.64.4.2 GetKey()	278
10.64.4.3 GetName()	278
10.64.4.4 GetNoOfItems()	278
10.64.4.5 GetSyngoDT()	279
10.64.4.6 GetValue() [1/2]	279
10.64.4.7 GetValue() [2/2]	279
10.64.4.8 GetVM()	279
10.64.4.9 GetVR()	279
10.64.4.10IsEmpty()	280
10.64.4.11operator<()	280
10.64.4.12operator=()	280
10.64.4.13operator==(())	280
10.64.4.14SetByteValue()	280
10.64.4.15SetKey()	281
10.64.4.16SetName()	281
10.64.4.17SetNoOfItems()	281

10.64.4.18SetSyngoDT()	281
10.64.4.19SetValue()	281
10.64.4.20SetVM()	281
10.64.4.21SetVR()	282
10.64.5 Friends And Related Function Documentation	282
10.64.5.1 operator<<	282
10.64.6 Member Data Documentation	282
10.64.6.1 DataField	282
10.64.6.2 KeyField	282
10.64.6.3 NameField	282
10.64.6.4 NoOfItemsField	283
10.64.6.5 SyngoDTField	283
10.64.6.6 ValueMultiplicityField	283
10.64.6.7 VRField	283
10.65gdcm::CSAHeader Class Reference	283
10.65.1 Detailed Description	285
10.65.2 Member Enumeration Documentation	285
10.65.2.1 CSAHeaderType	285
10.65.3 Constructor & Destructor Documentation	286
10.65.3.1 CSAHeader()	286
10.65.3.2 ~CSAHeader()	286
10.65.4 Member Function Documentation	286
10.65.4.1 FindCSAElementByName()	286
10.65.4.2 GetCSADataInfo()	286
10.65.4.3 GetCSAEEnd()	287
10.65.4.4 GetCSAElementByName()	287
10.65.4.5 GetCSAImageHeaderInfoTag()	287
10.65.4.6 GetCSASeriesHeaderInfoTag()	287



10.65.4.7 GetDataSet()	288
10.65.4.8 GetFormat()	288
10.65.4.9 GetInterfile()	288
10.65.4.10 GetMrProtocol()	288
10.65.4.11 LoadFromDataElement()	288
10.65.4.12 Print()	289
10.65.4.13 Read()	289
10.65.4.14 Write()	289
10.65.5 Friends And Related Function Documentation	289
10.65.5.1 operator<<	289
10.66gdcm::CSAHeaderDict Class Reference	290
10.66.1 Detailed Description	290
10.66.2 Member Typedef Documentation	290
10.66.2.1 ConstIterator	291
10.66.2.2 Iterator	291
10.66.2.3 MapCSAHeaderDictEntry	291
10.66.3 Constructor & Destructor Documentation	291
10.66.3.1 CSAHeaderDict()	291
10.66.4 Member Function Documentation	291
10.66.4.1 AddCSAHeaderDictEntry()	291
10.66.4.2 Begin()	291
10.66.4.3 End()	292
10.66.4.4 GetCSAHeaderDictEntry()	292
10.66.4.5 IsEmpty()	292
10.66.4.6 LoadDefault()	292
10.66.5 Friends And Related Function Documentation	292
10.66.5.1 Dicts	292
10.66.5.2 operator<<	292

10.67gdcm::CSAHeaderDictEntry Class Reference . . . . .	293
10.67.1 Detailed Description . . . . .	293
10.67.2 Constructor & Destructor Documentation . . . . .	294
10.67.2.1 CSAHeaderDictEntry() . . . . .	294
10.67.3 Member Function Documentation . . . . .	294
10.67.3.1 GetDescription() . . . . .	294
10.67.3.2 GetName() . . . . .	294
10.67.3.3 GetVM() . . . . .	294
10.67.3.4 GetVR() . . . . .	295
10.67.3.5 operator<() . . . . .	295
10.67.3.6 SetDescription() . . . . .	295
10.67.3.7 SetName() . . . . .	295
10.67.3.8 SetVM() . . . . .	295
10.67.3.9 SetVR() . . . . .	295
10.67.4 Friends And Related Function Documentation . . . . .	296
10.67.4.1 operator<< . . . . .	296
10.68gdcm::CSAHeaderDictException Class Reference . . . . .	296
10.69gdcm::network::CStoreRQ Class Reference . . . . .	297
10.69.1 Detailed Description . . . . .	298
10.69.2 Member Function Documentation . . . . .	298
10.69.2.1 ConstructPDV() . . . . .	298
10.70gdcm::network::CStoreRSP Class Reference . . . . .	298
10.70.1 Detailed Description . . . . .	299
10.70.2 Member Function Documentation . . . . .	299
10.70.2.1 ConstructPDV() . . . . .	299
10.71gdcm::Curve Class Reference . . . . .	300
10.71.1 Detailed Description . . . . .	301
10.71.2 Constructor & Destructor Documentation . . . . .	301

10.71.2.1 Curve() [1/2]	302
10.71.2.2 ~Curve()	302
10.71.2.3 Curve() [2/2]	302
10.71.3 Member Function Documentation	302
10.71.3.1 Decode()	302
10.71.3.2 GetAsPoints()	302
10.71.3.3 GetCurveDataDescriptor()	302
10.71.3.4 GetDataValueRepresentation()	303
10.71.3.5 GetDimensions()	303
10.71.3.6 GetGroup()	303
10.71.3.7 GetNumberOfCurves()	303
10.71.3.8 GetNumberOfPoints()	303
10.71.3.9 GetTypeInfoData()	303
10.71.3.10GetTypeInfoDataDescription()	303
10.71.3.11IsEmpty()	304
10.71.3.12Print()	304
10.71.3.13SetCoordinateStartValue()	304
10.71.3.14SetCoordinateStepValue()	304
10.71.3.15SetCurve()	304
10.71.3.16SetCurveDataDescriptor()	304
10.71.3.17SetCurveDescription()	305
10.71.3.18SetDataValueRepresentation()	305
10.71.3.19SetDimensions()	305
10.71.3.20SetGroup()	305
10.71.3.21SetNumberOfPoints()	305
10.71.3.22SetTypeInfoData()	305
10.71.3.23Update()	306
10.72gdcm::DataElement Class Reference	306

10.72.1 Detailed Description	309
10.72.2 Member Typedef Documentation	309
10.72.2.1 ValuePtr	309
10.72.3 Constructor & Destructor Documentation	309
10.72.3.1 DataElement() [1/2]	310
10.72.3.2 DataElement() [2/2]	310
10.72.4 Member Function Documentation	310
10.72.4.1 Clear()	310
10.72.4.2 Empty()	310
10.72.4.3 GetByteValue()	311
10.72.4.4 GetLength()	311
10.72.4.5 GetSequenceOfFragments() [1/2]	311
10.72.4.6 GetSequenceOfFragments() [2/2]	312
10.72.4.7 GetTag() [1/2]	312
10.72.4.8 GetTag() [2/2]	312
10.72.4.9 GetValue() [1/2]	312
10.72.4.10 GetValue() [2/2]	313
10.72.4.11 GetValueAsSQ()	313
10.72.4.12 GetVL() [1/2]	313
10.72.4.13 GetVL() [2/2]	313
10.72.4.14 GetVR()	314
10.72.4.15 IsEmpty()	314
10.72.4.16 IsUndefinedLength()	314
10.72.4.17 operator<()	315
10.72.4.18 operator=()	315
10.72.4.19 operator==(())	315
10.72.4.20 Read()	315
10.72.4.21 ReadOrSkip()	315

10.72.4.22	<a href="#">ReadPreValue()</a> . . . . .	316
10.72.4.23	<a href="#">ReadValue()</a> . . . . .	316
10.72.4.24	<a href="#">ReadValueWithLength()</a> . . . . .	316
10.72.4.25	<a href="#">ReadWithLength()</a> . . . . .	316
10.72.4.26	<a href="#">SetByteValue()</a> . . . . .	317
10.72.4.27	<a href="#">SetTag()</a> . . . . .	317
10.72.4.28	<a href="#">SetValue()</a> . . . . .	318
10.72.4.29	<a href="#">SetValueFieldLength()</a> . . . . .	318
10.72.4.30	<a href="#">SetVL()</a> . . . . .	318
10.72.4.31	<a href="#">SetVLToUndefined()</a> . . . . .	318
10.72.4.32	<a href="#">SetVR()</a> . . . . .	319
10.72.4.33	<a href="#">Write()</a> . . . . .	319
10.72.5	<a href="#">Friends And Related Function Documentation</a> . . . . .	319
10.72.5.1	<a href="#">operator&lt;&lt;</a> . . . . .	319
10.72.6	<a href="#">Member Data Documentation</a> . . . . .	319
10.72.6.1	<a href="#">TagField</a> . . . . .	320
10.72.6.2	<a href="#">ValueField</a> . . . . .	320
10.72.6.3	<a href="#">ValueLengthField</a> . . . . .	320
10.72.6.4	<a href="#">VRField</a> . . . . .	320
10.73	<a href="#">gdcm::DataElementException Class Reference</a> . . . . .	321
10.74	<a href="#">gdcm::DataEvent Class Reference</a> . . . . .	321
10.74.1	<a href="#">Detailed Description</a> . . . . .	323
10.74.2	<a href="#">Member Typedef Documentation</a> . . . . .	323
10.74.2.1	<a href="#">Self</a> . . . . .	323
10.74.2.2	<a href="#">Superclass</a> . . . . .	323
10.74.3	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	323
10.74.3.1	<a href="#">DataEvent()</a> <sup>[1/2]</sup> . . . . .	323
10.74.3.2	<a href="#">~DataEvent()</a> . . . . .	324

10.74.3.3 DataEvent() [2/2]	324
10.74.4 Member Function Documentation	324
10.74.4.1 CheckEvent()	324
10.74.4.2 GetData()	324
10.74.4.3 GetDataLength()	324
10.74.4.4 GetEventName()	324
10.74.4.5 MakeObject()	325
10.74.4.6 SetData()	325
10.75gdcm::DataSet Class Reference	325
10.75.1 Detailed Description	327
10.75.2 Member Typedef Documentation	328
10.75.2.1 ConstIterator	328
10.75.2.2 DataElementSet	328
10.75.2.3 Iterator	328
10.75.2.4 SizeType	328
10.75.3 Member Function Documentation	328
10.75.3.1 Begin() [1/2]	328
10.75.3.2 Begin() [2/2]	329
10.75.3.3 Clear()	329
10.75.3.4 ComputeDataElement()	329
10.75.3.5 ComputeGroupLength()	329
10.75.3.6 End() [1/2]	329
10.75.3.7 End() [2/2]	329
10.75.3.8 FindDataElement() [1/2]	330
10.75.3.9 FindDataElement() [2/2]	330
10.75.3.10FindNextDataElement()	330
10.75.3.11GetDataElement() [1/2]	331
10.75.3.12GetDataElement() [2/2]	331

10.75.3.13	<a href="#">GetDEEnd()</a>	331
10.75.3.14	<a href="#">GetDES()</a> [1/2]	331
10.75.3.15	<a href="#">GetDES()</a> [2/2]	332
10.75.3.16	<a href="#">GetLength()</a>	332
10.75.3.17	<a href="#">GetMediaStorage()</a>	332
10.75.3.18	<a href="#">GetPrivateCreator()</a>	332
10.75.3.19	<a href="#">Insert()</a>	332
10.75.3.20	<a href="#">InsertDataElement()</a>	333
10.75.3.21	<a href="#">IsEmpty()</a>	333
10.75.3.22	<a href="#">operator()()</a>	333
10.75.3.23	<a href="#">operator=()</a>	333
10.75.3.24	<a href="#">operator[]()</a>	333
10.75.3.25	<a href="#">Print()</a>	334
10.75.3.26	<a href="#">Read()</a>	334
10.75.3.27	<a href="#">ReadNested()</a>	334
10.75.3.28	<a href="#">ReadSelectedPrivateTags()</a>	334
10.75.3.29	<a href="#">ReadSelectedPrivateTagsWithLength()</a>	334
10.75.3.30	<a href="#">ReadSelectedTags()</a>	335
10.75.3.31	<a href="#">ReadSelectedTagsWithLength()</a>	335
10.75.3.32	<a href="#">ReadUpToTag()</a>	335
10.75.3.33	<a href="#">ReadUpToTagWithLength()</a>	335
10.75.3.34	<a href="#">ReadWithLength()</a>	335
10.75.3.35	<a href="#">Remove()</a>	336
10.75.3.36	<a href="#">Replace()</a>	336
10.75.3.37	<a href="#">ReplaceEmpty()</a>	336
10.75.3.38	<a href="#">Size()</a>	337
10.75.3.39	<a href="#">Write()</a>	337
10.75.4	<a href="#">Friends And Related Function Documentation</a>	337

10.75.4.1 CSAHeader . . . . .	337
10.75.4.2 operator<< . . . . .	337
10.76gdcM::DataSetEvent Class Reference . . . . .	338
10.76.1 Detailed Description . . . . .	339
10.76.2 Member Typedef Documentation . . . . .	339
10.76.2.1 Self . . . . .	339
10.76.2.2 Superclass . . . . .	339
10.76.3 Constructor & Destructor Documentation . . . . .	339
10.76.3.1 DataSetEvent() [1/2] . . . . .	339
10.76.3.2 ~DataSetEvent() . . . . .	340
10.76.3.3 DataSetEvent() [2/2] . . . . .	340
10.76.4 Member Function Documentation . . . . .	340
10.76.4.1 CheckEvent() . . . . .	340
10.76.4.2 GetDataSet() . . . . .	340
10.76.4.3 GetEventName() . . . . .	340
10.76.4.4 MakeObject() . . . . .	341
10.77gdcM::DataSetHelper Class Reference . . . . .	341
10.77.1 Detailed Description . . . . .	341
10.77.2 Member Function Documentation . . . . .	341
10.77.2.1 ComputeVR() . . . . .	341
10.78gdcM::Decoder Class Reference . . . . .	342
10.78.1 Detailed Description . . . . .	342
10.78.2 Constructor & Destructor Documentation . . . . .	342
10.78.2.1 ~Decoder() . . . . .	343
10.78.3 Member Function Documentation . . . . .	343
10.78.3.1 CanDecode() . . . . .	343
10.78.3.2 Decode() . . . . .	343
10.78.3.3 DecodeByStreams() . . . . .	343



10.79gdcm::DefinedTerms Class Reference . . . . .	344
10.79.1 Detailed Description . . . . .	344
10.79.2 Constructor & Destructor Documentation . . . . .	344
10.79.2.1 DefinedTerms() . . . . .	344
10.80gdcm::Defs Class Reference . . . . .	344
10.80.1 Detailed Description . . . . .	345
10.80.2 Constructor & Destructor Documentation . . . . .	345
10.80.2.1 Defs() . . . . .	346
10.80.2.2 ~Defs() . . . . .	346
10.80.3 Member Function Documentation . . . . .	346
10.80.3.1 GetIODFromFile() . . . . .	346
10.80.3.2 GetIODNameFromMediaStorage() . . . . .	346
10.80.3.3 GetIODs() [1/2] . . . . .	346
10.80.3.4 GetIODs() [2/2] . . . . .	347
10.80.3.5 GetMacros() [1/2] . . . . .	347
10.80.3.6 GetMacros() [2/2] . . . . .	347
10.80.3.7 GetModules() [1/2] . . . . .	347
10.80.3.8 GetModules() [2/2] . . . . .	347
10.80.3.9 GetTypeFromTag() . . . . .	348
10.80.3.10IsEmpty() . . . . .	348
10.80.3.11LoadDefaults() . . . . .	348
10.80.3.12LoadFromFile() . . . . .	348
10.80.3.13Verify() [1/2] . . . . .	348
10.80.3.14Verify() [2/2] . . . . .	348
10.80.4 Friends And Related Function Documentation . . . . .	348
10.80.4.1 Global . . . . .	349
10.81gdcm::DeltaEncodingCodec Class Reference . . . . .	349
10.81.1 Detailed Description . . . . .	350

10.81.2 Constructor & Destructor Documentation . . . . .	350
10.81.2.1 DeltaEncodingCodec() . . . . .	350
10.81.2.2 ~DeltaEncodingCodec() . . . . .	350
10.81.3 Member Function Documentation . . . . .	350
10.81.3.1 CanDecode() . . . . .	350
10.81.3.2 Decode() [1/2] . . . . .	351
10.81.3.3 Decode() [2/2] . . . . .	351
10.82gdcm::DICOMDIR Class Reference . . . . .	351
10.82.1 Detailed Description . . . . .	351
10.82.2 Constructor & Destructor Documentation . . . . .	351
10.82.2.1 DICOMDIR() [1/2] . . . . .	352
10.82.2.2 DICOMDIR() [2/2] . . . . .	352
10.83gdcm::DICOMDIRGenerator Class Reference . . . . .	352
10.83.1 Detailed Description . . . . .	353
10.83.2 Member Typedef Documentation . . . . .	353
10.83.2.1 FilenamesType . . . . .	353
10.83.2.2 FilenameType . . . . .	354
10.83.3 Constructor & Destructor Documentation . . . . .	354
10.83.3.1 DICOMDIRGenerator() . . . . .	354
10.83.3.2 ~DICOMDIRGenerator() . . . . .	354
10.83.4 Member Function Documentation . . . . .	354
10.83.4.1 AddImageDirectoryRecord() . . . . .	354
10.83.4.2 AddPatientDirectoryRecord() . . . . .	354
10.83.4.3 AddSeriesDirectoryRecord() . . . . .	354
10.83.4.4 AddStudyDirectoryRecord() . . . . .	355
10.83.4.5 Generate() . . . . .	355
10.83.4.6 GetFile() . . . . .	355
10.83.4.7 GetScanner() . . . . .	355

10.83.4.8 SetDescriptor()	355
10.83.4.9 SetFile()	355
10.83.4.10SetFileNames()	356
10.83.4.11SetRootDirectory()	356
10.84gdcmm::Dict Class Reference	356
10.84.1 Detailed Description	357
10.84.2 Member Typedef Documentation	357
10.84.2.1 ConstIterator	357
10.84.2.2 Iterator	357
10.84.2.3 MapDictEntry	357
10.84.3 Constructor & Destructor Documentation	358
10.84.3.1 Dict()	358
10.84.4 Member Function Documentation	358
10.84.4.1 AddDictEntry()	358
10.84.4.2 Begin()	358
10.84.4.3 End()	358
10.84.4.4 GetDictEntry()	359
10.84.4.5 GetDictEntryByKeyword()	359
10.84.4.6 GetDictEntryByName()	359
10.84.4.7 GetKeywordFromTag()	359
10.84.4.8 IsEmpty()	360
10.84.4.9 LoadDefault()	360
10.84.5 Friends And Related Function Documentation	360
10.84.5.1 Dicts	360
10.84.5.2 operator<<	360
10.85gdcmm::DictConverter Class Reference	360
10.85.1 Detailed Description	361
10.85.2 Member Enumeration Documentation	361

10.85.2.1 OutputTypes . . . . .	361
10.85.3 Constructor & Destructor Documentation . . . . .	362
10.85.3.1 DictConverter() . . . . .	362
10.85.3.2 ~DictConverter() . . . . .	362
10.85.4 Member Function Documentation . . . . .	362
10.85.4.1 AddGroupLength() . . . . .	362
10.85.4.2 Convert() . . . . .	362
10.85.4.3 ConvertToCXX() . . . . .	362
10.85.4.4 ConvertToXML() . . . . .	363
10.85.4.5 GetDictName() . . . . .	363
10.85.4.6 GetInputFilename() . . . . .	363
10.85.4.7 GetOutputFilename() . . . . .	363
10.85.4.8 GetOutputType() . . . . .	363
10.85.4.9 Readuint16() . . . . .	363
10.85.4.10ReadVM() . . . . .	364
10.85.4.11ReadVR() . . . . .	364
10.85.4.12SetDictName() . . . . .	364
10.85.4.13SetInputFileName() . . . . .	364
10.85.4.14SetOutputFileName() . . . . .	364
10.85.4.15SetOutputType() . . . . .	364
10.85.4.16WriteFooter() . . . . .	365
10.85.4.17WriteHeader() . . . . .	365
10.86gdcmm::DictEntry Class Reference . . . . .	365
10.86.1 Detailed Description . . . . .	366
10.86.2 Constructor & Destructor Documentation . . . . .	366
10.86.2.1 DictEntry() . . . . .	366
10.86.3 Member Function Documentation . . . . .	366
10.86.3.1 GetKeyword() . . . . .	367

10.86.3.2	GetName()	367
10.86.3.3	GetRetired()	367
10.86.3.4	GetVM()	367
10.86.3.5	GetVR()	368
10.86.3.6	IsUnique()	368
10.86.3.7	SetElementXX()	368
10.86.3.8	SetGroupXX()	368
10.86.3.9	SetKeyword()	368
10.86.3.10	SetName()	369
10.86.3.11	SetRetired()	369
10.86.3.12	SetVM()	369
10.86.3.13	SetVR()	369
10.86.4	Friends And Related Function Documentation	369
10.86.4.1	Dict	369
10.86.4.2	operator<<	369
10.87	gdcmm::DictPrinter Class Reference	370
10.87.1	Detailed Description	371
10.87.2	Constructor & Destructor Documentation	371
10.87.2.1	DictPrinter()	371
10.87.2.2	~DictPrinter()	371
10.87.3	Member Function Documentation	371
10.87.3.1	Print()	371
10.87.3.2	PrintDataElement2()	372
10.87.3.3	PrintDataSet2()	372
10.88	gdcmm::Dicts Class Reference	372
10.88.1	Detailed Description	373
10.88.2	Member Enumeration Documentation	373
10.88.2.1	ConstructorType	373

10.88.3 Constructor & Destructor Documentation . . . . .	373
10.88.3.1 Dicts() . . . . .	374
10.88.3.2 ~Dicts() . . . . .	374
10.88.4 Member Function Documentation . . . . .	374
10.88.4.1 GetConstructorString() . . . . .	374
10.88.4.2 GetCSAHeaderDict() . . . . .	374
10.88.4.3 GetDictEntry() [1/2] . . . . .	374
10.88.4.4 GetDictEntry() [2/2] . . . . .	375
10.88.4.5 GetPrivateDict() [1/2] . . . . .	375
10.88.4.6 GetPrivateDict() [2/2] . . . . .	375
10.88.4.7 GetPublicDict() . . . . .	375
10.88.4.8 IsEmpty() . . . . .	375
10.88.4.9 LoadDefaults() . . . . .	375
10.88.5 Friends And Related Function Documentation . . . . .	376
10.88.5.1 Global . . . . .	376
10.88.5.2 operator<< . . . . .	376
10.89gdcm::network::DIMSE Class Reference . . . . .	376
10.89.1 Detailed Description . . . . .	377
10.89.2 Member Enumeration Documentation . . . . .	377
10.89.2.1 CommandTypes . . . . .	377
10.90gdcm::DirectionCosines Class Reference . . . . .	378
10.90.1 Detailed Description . . . . .	378
10.90.2 Constructor & Destructor Documentation . . . . .	379
10.90.2.1 DirectionCosines() [1/2] . . . . .	379
10.90.2.2 DirectionCosines() [2/2] . . . . .	379
10.90.2.3 ~DirectionCosines() . . . . .	379
10.90.3 Member Function Documentation . . . . .	379
10.90.3.1 ComputeDistAlongNormal() . . . . .	379

10.90.3.2 Cross()	379
10.90.3.3 CrossDot()	380
10.90.3.4 Dot() [1/2]	380
10.90.3.5 Dot() [2/2]	380
10.90.3.6 IsValid()	380
10.90.3.7 Normalize() [1/2]	380
10.90.3.8 Normalize() [2/2]	381
10.90.3.9 operator const double *()	381
10.90.3.10 Print()	381
10.90.3.11 SetFromString()	381
10.91gdcmm::Directory Class Reference	381
10.91.1 Detailed Description	382
10.91.2 Member Typedef Documentation	383
10.91.2.1 FilenamesType	383
10.91.2.2 FilenameType	383
10.91.3 Constructor & Destructor Documentation	383
10.91.3.1 Directory()	383
10.91.3.2 ~Directory()	383
10.91.4 Member Function Documentation	383
10.91.4.1 Explore()	384
10.91.4.2 GetDirectories()	384
10.91.4.3 GetFilenames()	384
10.91.4.4 GetToplevel()	384
10.91.4.5 Load()	385
10.91.4.6 Print()	385
10.91.5 Friends And Related Function Documentation	385
10.91.5.1 operator<<	385
10.92gdcmm::DirectoryHelper Class Reference	386

10.92.1 Detailed Description . . . . .	386
10.92.2 Member Function Documentation . . . . .	386
10.92.2.1 GetCTImageSeriesUIDs() . . . . .	386
10.92.2.2 GetFileNamesFromSeriesUIDs() . . . . .	387
10.92.2.3 GetFrameOfReference() . . . . .	387
10.92.2.4 GetMRImageSeriesUIDs() . . . . .	387
10.92.2.5 GetRTStructSeriesUIDs() . . . . .	387
10.92.2.6 GetSeriesUIDsBySOPClassUID() . . . . .	387
10.92.2.7 GetSOPClassUID() . . . . .	388
10.92.2.8 GetStringValueFromTag() . . . . .	388
10.92.2.9 LoadImageFromFiles() . . . . .	388
10.92.2.10RetrieveSOPInstanceUIDFromIndex() . . . . .	388
10.92.2.11RetrieveSOPInstanceUIDFromZPosition() . . . . .	388
10.93gdcm::DummyValueGenerator Class Reference . . . . .	388
10.93.1 Detailed Description . . . . .	389
10.93.2 Member Function Documentation . . . . .	389
10.93.2.1 Generate() . . . . .	389
10.94gdcm::Dumper Class Reference . . . . .	389
10.94.1 Detailed Description . . . . .	390
10.94.2 Constructor & Destructor Documentation . . . . .	390
10.94.2.1 Dumper() . . . . .	391
10.94.2.2 ~Dumper() . . . . .	391
10.95gdcm::Element< TVR, TVM > Class Template Reference . . . . .	391
10.95.1 Detailed Description . . . . .	393
10.95.2 Member Typedef Documentation . . . . .	393
10.95.2.1 Type . . . . .	393
10.95.3 Member Function Documentation . . . . .	393
10.95.3.1 GetAsDataElement() . . . . .	393



10.95.3.2 GetLength()	394
10.95.3.3 GetValue() [1/2]	394
10.95.3.4 GetValue() [2/2]	394
10.95.3.5 GetValues()	394
10.95.3.6 GetVM()	394
10.95.3.7 GetVR()	394
10.95.3.8 operator[]()	395
10.95.3.9 Print()	395
10.95.3.10 Read()	395
10.95.3.11 Set()	395
10.95.3.12 SetFromDataElement()	395
10.95.3.13 SetNoSwap()	395
10.95.3.14 SetValue()	396
10.95.3.15 Write()	396
10.95.4 Member Data Documentation	396
10.95.4.1 Internal	396
10.96gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	396
10.96.1 Member Typedef Documentation	397
10.96.1.1 Parent	397
10.96.2 Member Function Documentation	397
10.96.2.1 SetLength()	398
10.97gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	398
10.97.1 Member Typedef Documentation	399
10.97.1.1 Type	399
10.97.2 Constructor & Destructor Documentation	399
10.97.2.1 Element() [1/2]	399
10.97.2.2 ~Element()	399
10.97.2.3 Element() [2/2]	399

10.97.3 Member Function Documentation . . . . .	400
10.97.3.1 GetAsDataElement() . . . . .	400
10.97.3.2 GetLength() . . . . .	400
10.97.3.3 GetValue() [1/2] . . . . .	400
10.97.3.4 GetValue() [2/2] . . . . .	400
10.97.3.5 GetVM() . . . . .	400
10.97.3.6 GetVR() . . . . .	401
10.97.3.7 operator=() . . . . .	401
10.97.3.8 operator[]() . . . . .	401
10.97.3.9 Print() . . . . .	401
10.97.3.10 Read() . . . . .	401
10.97.3.11 Set() . . . . .	402
10.97.3.12 SetArray() . . . . .	402
10.97.3.13 SetFromDataElement() . . . . .	402
10.97.3.14 SetLength() . . . . .	402
10.97.3.15 SetNoSwap() . . . . .	402
10.97.3.16 SetValue() . . . . .	403
10.97.3.17 Write() . . . . .	403
10.97.3.18 WriteASCII() . . . . .	403
10.98 gdcmm::Element< TVR, VM::VM2_2n > Class Template Reference . . . . .	403
10.98.1 Member Typedef Documentation . . . . .	404
10.98.1.1 Parent . . . . .	404
10.98.2 Member Function Documentation . . . . .	404
10.98.2.1 SetLength() . . . . .	405
10.99 gdcmm::Element< TVR, VM::VM2_n > Class Template Reference . . . . .	405
10.99.1 Member Typedef Documentation . . . . .	406
10.99.1.1 Parent . . . . .	406
10.99.2 Member Function Documentation . . . . .	406

10.99.2.1 SetLength()	407
10.100dcm::Element< TVR, VM::VM3_3n > Class Template Reference	407
10.100.1Member Typedef Documentation	408
10.100.1.1Parent	408
10.100.2Member Function Documentation	408
10.100.2.1SetLength()	409
10.101dcm::Element< TVR, VM::VM3_n > Class Template Reference	409
10.101.1Member Typedef Documentation	410
10.101.1.1Parent	410
10.101.2Member Function Documentation	410
10.101.2.1SetLength()	411
10.102dcm::Element< VR::AS, VM::VM5 > Class Template Reference	411
10.102.1Member Function Documentation	411
10.102.1.1GetLength()	411
10.102.1.2Print()	411
10.102.2Member Data Documentation	411
10.102.2.1Internal	412
10.103dcm::Element< VR::OB, VM::VM1 > Class Template Reference	412
10.104dcm::Element< VR::OW, VM::VM1 > Class Template Reference	413
10.105dcm::ElementDisableCombinations< TVR, TVM > Class Template Reference	415
10.105.1Detailed Description	416
10.106dcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference	416
10.107dcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference	416
10.108dcm::EmptyMaskGenerator Class Reference	416
10.108.1Detailed Description	417
10.108.2Member Enumeration Documentation	418
10.108.2.1ISOPClassUIDMode	418
10.108.3Constructor & Destructor Documentation	418

10.108.3.1EmptyMaskGenerator()	418
10.108.3.2~EmptyMaskGenerator()	418
10.108.4Member Function Documentation	418
10.108.4.1Execute()	418
10.108.4.2SetInputDirectory()	419
10.108.4.3SetOutputDirectory()	419
10.108.4.4SetSOPClassUIDMode()	419
10.109dcm::EncapsulatedDocument Class Reference	419
10.109.1Detailed Description	420
10.109.2Constructor & Destructor Documentation	420
10.109.2.1EncapsulatedDocument()	420
10.110dcm::EncodingImplementation< T > Class Template Reference	420
10.110.1Detailed Description	420
10.111dcm::EncodingImplementation< VR::VRASCII > Class Template Reference	421
10.111.1Member Function Documentation	421
10.111.1.1Read()	421
10.111.1.2ReadComputeLength()	421
10.111.1.3ReadNoSwap()	422
10.111.1.4Write() [1/3]	422
10.111.1.5Write() [2/3]	422
10.111.1.6Write() [3/3]	422
10.112dcm::EncodingImplementation< VR::VRBINARY > Class Template Reference	423
10.112.1Member Function Documentation	423
10.112.1.1Read()	423
10.112.1.2ReadComputeLength()	423
10.112.1.3ReadNoSwap()	423
10.112.1.4Write()	424
10.113dcm::EndEvent Class Reference	424

10.114	dcm::EnumeratedValues Class Reference . . . . .	425
10.114.1	Detailed Description . . . . .	425
10.114.2	Constructor & Destructor Documentation . . . . .	426
10.114.2.1	EnumeratedValues() . . . . .	426
10.115	dcm::Event Class Reference . . . . .	426
10.115.1	Detailed Description . . . . .	428
10.115.2	Constructor & Destructor Documentation . . . . .	428
10.115.2.1	Event() [1/2] . . . . .	428
10.115.2.2	Event() [2/2] . . . . .	428
10.115.2.3	~Event() . . . . .	428
10.115.3	Member Function Documentation . . . . .	428
10.115.3.1	CheckEvent() . . . . .	428
10.115.3.2	GetEventName() . . . . .	429
10.115.3.3	MakeObject() . . . . .	429
10.115.3.4	Print() . . . . .	429
10.116	dcm::Exception Class Reference . . . . .	430
10.116.1	Detailed Description . . . . .	431
10.116.2	Constructor & Destructor Documentation . . . . .	431
10.116.2.1	Exception() . . . . .	431
10.116.2.2	~Exception() . . . . .	431
10.116.3	Member Function Documentation . . . . .	431
10.116.3.1	GetDescription() . . . . .	431
10.116.3.2	what() . . . . .	432
10.117	dcm::ExitEvent Class Reference . . . . .	432
10.118	dcm::ExplicitDataElement Class Reference . . . . .	433
10.118.1	Detailed Description . . . . .	434
10.118.2	Member Function Documentation . . . . .	435
10.118.2.1	GetLength() . . . . .	435

10.118.2.2	<a href="#">Read()</a>	435
10.118.2.3	<a href="#">ReadPreValue()</a>	435
10.118.2.4	<a href="#">ReadValue()</a>	435
10.118.2.5	<a href="#">ReadWithLength()</a>	435
10.118.2.6	<a href="#">Write()</a>	436
10.119	<a href="#">dcm::ExplicitImplicitDataElement Class Reference</a>	436
10.119.1	<a href="#">Detailed Description</a>	437
10.119.2	<a href="#">Member Function Documentation</a>	438
10.119.2.1	<a href="#">GetLength()</a>	438
10.119.2.2	<a href="#">Read()</a>	438
10.119.2.3	<a href="#">ReadPreValue()</a>	438
10.119.2.4	<a href="#">ReadValue()</a>	438
10.119.2.5	<a href="#">ReadWithLength()</a>	438
10.120	<a href="#">dcm::Fiducials Class Reference</a>	439
10.120.1	<a href="#">Detailed Description</a>	439
10.120.2	<a href="#">Constructor &amp; Destructor Documentation</a>	439
10.120.2.1	<a href="#">Fiducials()</a>	439
10.121	<a href="#">dcm::File Class Reference</a>	439
10.121.1	<a href="#">Detailed Description</a>	441
10.121.2	<a href="#">Constructor &amp; Destructor Documentation</a>	441
10.121.2.1	<a href="#">File()</a>	441
10.121.2.2	<a href="#">~File()</a>	442
10.121.3	<a href="#">Member Function Documentation</a>	442
10.121.3.1	<a href="#">GetDataSet()</a> [1/2]	442
10.121.3.2	<a href="#">GetDataSet()</a> [2/2]	442
10.121.3.3	<a href="#">GetHeader()</a> [1/2]	442
10.121.3.4	<a href="#">GetHeader()</a> [2/2]	443
10.121.3.5	<a href="#">Read()</a>	443

10.121.3.6	DataSet()	443
10.121.3.7	SetHeader()	443
10.121.3.8	Write()	443
10.121.4	Friends And Related Function Documentation	443
10.121.4.1	operator<<	444
10.122	dcm::FileAnonymizer Class Reference	444
10.122.1	Detailed Description	445
10.122.2	Constructor & Destructor Documentation	446
10.122.2.1	FileAnonymizer()	446
10.122.2.2	~FileAnonymizer()	446
10.122.3	Member Function Documentation	446
10.122.3.1	Empty()	446
10.122.3.2	Remove()	447
10.122.3.3	Replace() [1/2]	447
10.122.3.4	Replace() [2/2]	447
10.122.3.5	SetInputFileName()	447
10.122.3.6	SetOutputFileName()	448
10.122.3.7	Write()	448
10.123	dcm::FileChangeTransferSyntax Class Reference	448
10.123.1	Detailed Description	449
10.123.2	Constructor & Destructor Documentation	450
10.123.2.1	FileChangeTransferSyntax()	450
10.123.2.2	~FileChangeTransferSyntax()	450
10.123.3	Member Function Documentation	450
10.123.3.1	Change()	450
10.123.3.2	GetCodec()	450
10.123.3.3	New()	451
10.123.3.4	SetInputFileName()	451

10.123.3.5SetOutputFileName()	451
10.123.3.6SetTransferSyntax()	451
10.124dcm::FileDecompressLookupTable Class Reference	452
10.124.1Detailed Description	453
10.124.2Constructor & Destructor Documentation	453
10.124.2.1FileDecompressLookupTable()	453
10.124.2.2~FileDecompressLookupTable()	453
10.124.3Member Function Documentation	453
10.124.3.1Change()	454
10.124.3.2GetFile()	454
10.124.3.3GetPixmap() [1/2]	454
10.124.3.4GetPixmap() [2/2]	454
10.124.3.5SetFile()	454
10.124.3.6SetPixmap()	454
10.125dcm::FileDerivation Class Reference	455
10.125.1Detailed Description	455
10.125.2Constructor & Destructor Documentation	456
10.125.2.1FileDerivation()	456
10.125.2.2~FileDerivation()	456
10.125.3Member Function Documentation	456
10.125.3.1AddDerivationDescription()	456
10.125.3.2AddPurposeOfReferenceCodeSequence()	456
10.125.3.3AddReference()	456
10.125.3.4AddSourceImageSequence()	457
10.125.3.5Derive()	457
10.125.3.6GetFile() [1/2]	457
10.125.3.7GetFile() [2/2]	457
10.125.3.8SetAppendDerivationHistory()	457



10.125.3.9SetDerivationCodeSequenceCodeValue()	458
10.125.3.10SetDerivationDescription()	458
10.125.3.11SetFile()	458
10.125.3.12SetPurposeOfReferenceCodeSequenceCodeValue()	458
10.126.0dcm::FileExplicitFilter Class Reference	459
10.126.1Detailed Description	459
10.126.2Constructor & Destructor Documentation	460
10.126.2.1FileExplicitFilter()	460
10.126.2.2~FileExplicitFilter()	460
10.126.3Member Function Documentation	460
10.126.3.1Change()	460
10.126.3.2ChangeFMI()	460
10.126.3.3GetFile()	460
10.126.3.4ProcessDataSet()	461
10.126.3.5SetChangePrivateTags()	461
10.126.3.6SetFile()	461
10.126.3.7SetRecomputeItemLength()	461
10.126.3.8SetRecomputeSequenceLength()	461
10.126.3.9SetUseVRUN()	462
10.127.0dcm::FileMetaInformation Class Reference	462
10.127.1Detailed Description	464
10.127.2Constructor & Destructor Documentation	464
10.127.2.1FileMetaInformation() [1/2]	465
10.127.2.2~FileMetaInformation()	465
10.127.2.3FileMetaInformation() [2/2]	465
10.127.3Member Function Documentation	465
10.127.3.1AppendImplementationClassUID()	465
10.127.3.2ComputeDataSetMediaStorageSOPClass()	465

10.127.3.3ComputeDataSetTransferSyntax()	465
10.127.3.4Default()	466
10.127.3.5FillFromDataSet()	466
10.127.3.6GetDataSetTransferSyntax()	466
10.127.3.7GetFileMetaInformationVersion()	466
10.127.3.8GetFullLength()	466
10.127.3.9GetGDCMImplementationClassUID()	466
10.127.3.10GetGDCMImplementationVersionName()	467
10.127.3.10GetGDCMSourceApplicationEntityTitle()	467
10.127.3.10GetImplementationClassUID()	467
10.127.3.10GetImplementationVersionName()	467
10.127.3.10GetMediaStorage()	467
10.127.3.10GetMediaStorageAsString()	467
10.127.3.10GetMetaInformationTS()	467
10.127.3.10GetPreamble() [1/2]	468
10.127.3.10GetPreamble() [2/2]	468
10.127.3.10GetSourceApplicationEntityTitle()	468
10.127.3.20Insert()	468
10.127.3.21IsValid()	468
10.127.3.22Read()	468
10.127.3.23ReadCompat()	469
10.127.3.24ReadCompatInternal()	469
10.127.3.25Replace()	469
10.127.3.26SetDataSetTransferSyntax()	469
10.127.3.27SetImplementationClassUID()	470
10.127.3.28SetImplementationVersionName()	470
10.127.3.29SetPreamble()	470
10.127.3.30SetSourceApplicationEntityTitle()	470

10.127.3.3Write()	470
10.127.4Friends And Related Function Documentation	470
10.127.4.1operator<<	471
10.127.5Member Data Documentation	471
10.127.5.1DataSetMS	471
10.127.5.2DataSetTS	471
10.127.5.3MetaInformationTS	471
10.128dcm::Filename Class Reference	471
10.128.1Detailed Description	472
10.128.2Constructor & Destructor Documentation	472
10.128.2.1Filename()	472
10.128.3Member Function Documentation	473
10.128.3.1EndWith()	473
10.128.3.2GetExtension()	473
10.128.3.3GetFileName()	473
10.128.3.4GetName()	473
10.128.3.5GetPath()	473
10.128.3.6IsEmpty()	474
10.128.3.7IsIdentical()	474
10.128.3.8Join()	474
10.128.3.9operator const char *()	474
10.128.3.10ToUnixSlashes()	474
10.128.3.11ToWindowsSlashes()	475
10.129dcm::FileNameEvent Class Reference	475
10.129.1Detailed Description	476
10.129.2Member Typedef Documentation	477
10.129.2.1Self	477
10.129.2.2Superclass	477

10.129.3	Constructor & Destructor Documentation	477
10.129.3.1	FileNameEvent() [1/2]	477
10.129.3.2	~FileNameEvent()	477
10.129.3.3	FileNameEvent() [2/2]	477
10.129.4	Member Function Documentation	477
10.129.4.1	CheckEvent()	478
10.129.4.2	GetEventName()	478
10.129.4.3	GetFileName()	478
10.129.4.4	MakeObject()	478
10.129.4.5	SetFileName()	478
10.130	dcm::FilenameGenerator Class Reference	479
10.130.1	Detailed Description	479
10.130.2	Member Typedef Documentation	480
10.130.2.1	FileNamesType	480
10.130.2.2	FilenameType	480
10.130.2.3	SizeType	480
10.130.3	Constructor & Destructor Documentation	480
10.130.3.1	FilenameGenerator()	480
10.130.3.2	~FilenameGenerator()	480
10.130.4	Member Function Documentation	480
10.130.4.1	Generate()	481
10.130.4.2	GetFilename()	481
10.130.4.3	GetFileNames()	481
10.130.4.4	GetNumberOfFileNames()	481
10.130.4.5	GetPattern()	482
10.130.4.6	GetPrefix()	482
10.130.4.7	SetNumberOfFileNames()	482
10.130.4.8	SetPattern()	482

10.130.4.9SetPrefix()	482
10.131dcm::FileSet Class Reference	483
10.131.1Detailed Description	483
10.131.2Member Typedef Documentation	483
10.131.2.1FileType	483
10.131.2.2FileType	483
10.131.3Constructor & Destructor Documentation	484
10.131.3.1FileSet()	484
10.131.4Member Function Documentation	484
10.131.4.1AddFile() [1/2]	484
10.131.4.2AddFile() [2/2]	484
10.131.4.3GetFiles()	484
10.131.4.4SetFiles()	484
10.131.5Friends And Related Function Documentation	485
10.131.5.1operator<<	485
10.132dcm::FileStreamer Class Reference	485
10.132.1Detailed Description	487
10.132.2Constructor & Destructor Documentation	487
10.132.2.1FileStreamer()	487
10.132.2.2~FileStreamer()	487
10.132.3Member Function Documentation	487
10.132.3.1AppendToDataElement()	487
10.132.3.2AppendToGroupDataElement()	488
10.132.3.3CheckDataElement()	488
10.132.3.4CheckTemplateFileName()	488
10.132.3.5New()	488
10.132.3.6ReserveDataElement()	488
10.132.3.7ReserveGroupDataElement()	489

10.132.3.8SetOutputFileName()	489
10.132.3.9SetTemplateFileName()	489
10.132.3.10StartDataElement()	489
10.132.3.11StartGroupDataElement()	489
10.132.3.12StopDataElement()	490
10.132.3.13StopGroupDataElement()	490
10.133dcm::FileWithName Class Reference	490
10.133.1Detailed Description	491
10.133.2Constructor & Destructor Documentation	491
10.133.2.1FileWithName()	492
10.133.3Member Data Documentation	492
10.133.3.1filename	492
10.134dcm::FindPatientRootQuery Class Reference	492
10.134.1Detailed Description	493
10.134.2Constructor & Destructor Documentation	493
10.134.2.1FindPatientRootQuery()	493
10.134.3Member Function Documentation	494
10.134.3.1GetAbstractSyntaxUID()	494
10.134.3.2GetTagListByLevel()	494
10.134.3.3InitializeDataSet()	494
10.134.3.4ValidateQuery()	494
10.134.4Friends And Related Function Documentation	495
10.134.4.1QueryFactory	495
10.135dcm::FindStudyRootQuery Class Reference	495
10.135.1Detailed Description	496
10.135.2Constructor & Destructor Documentation	496
10.135.2.1FindStudyRootQuery()	496
10.135.3Member Function Documentation	497

10.135.3.1GetAbstractSyntaxUID()	497
10.135.3.2GetTagListByLevel()	497
10.135.3.3InitializeDataSet()	497
10.135.3.4ValidateQuery()	497
10.135.4Friends And Related Function Documentation	498
10.135.4.1QueryFactory	498
10.136dcm::Fragment Class Reference	498
10.136.1Detailed Description	499
10.136.2Constructor & Destructor Documentation	500
10.136.2.1Fragment()	500
10.136.3Member Function Documentation	500
10.136.3.1ComputeLength()	500
10.136.3.2GetLength()	500
10.136.3.3Read()	500
10.136.3.4ReadBacktrack()	500
10.136.3.5ReadPreValue()	501
10.136.3.6ReadValue()	501
10.136.3.7Write()	501
10.136.4Friends And Related Function Documentation	501
10.136.4.1operator<<	501
10.137dcm::Global Class Reference	501
10.137.1Detailed Description	502
10.137.2Constructor & Destructor Documentation	502
10.137.2.1Global()	503
10.137.2.2~Global()	503
10.137.3Member Function Documentation	503
10.137.3.1Append()	503
10.137.3.2GetDefs()	503

10.137.3.3GetDicts() [1/2]	504
10.137.3.4GetDicts() [2/2]	504
10.137.3.5GetInstance()	504
10.137.3.6LoadResourcesFiles()	504
10.137.3.7Locate()	505
10.137.3.8Prepend()	505
10.137.4Friends And Related Function Documentation	505
10.137.4.1operator<<	505
10.138dcm::GroupDict Class Reference	505
10.138.1Detailed Description	506
10.138.2Member Typedef Documentation	506
10.138.2.1GroupStringVector	506
10.138.3Constructor & Destructor Documentation	506
10.138.3.1GroupDict()	506
10.138.3.2~GroupDict()	507
10.138.4Member Function Documentation	507
10.138.4.1Add()	507
10.138.4.2GetAbbreviation()	507
10.138.4.3GetName()	507
10.138.4.4Insert()	507
10.138.4.5Size()	508
10.138.5Friends And Related Function Documentation	508
10.138.5.1operator<<	508
10.139dcm::IconImageFilter Class Reference	508
10.139.1Detailed Description	509
10.139.2Constructor & Destructor Documentation	509
10.139.2.1IconImageFilter()	509
10.139.2.2~IconImageFilter()	509



10.139.3	Member Function Documentation	510
10.139.3.1	Extract()	510
10.139.3.2	ExtractIconImages()	510
10.139.3.3	ExtractVeprolIconImages()	510
10.139.3.4	GetFile() [1/2]	510
10.139.3.5	GetFile() [2/2]	510
10.139.3.6	GetIconImage()	511
10.139.3.7	GetNumberOfIconImages()	511
10.139.3.8	SetFile()	511
10.140	dcm::IconImageGenerator Class Reference	511
10.140.1	Detailed Description	512
10.140.2	Constructor & Destructor Documentation	512
10.140.2.1	IconImageGenerator()	513
10.140.2.2	~IconImageGenerator()	513
10.140.3	Member Function Documentation	513
10.140.3.1	AutoPixelMinMax()	513
10.140.3.2	ConvertRGBToPaletteColor()	513
10.140.3.3	Generate()	513
10.140.3.4	GetIconImage()	514
10.140.3.5	GetPixmap() [1/2]	514
10.140.3.6	GetPixmap() [2/2]	514
10.140.3.7	SetOutputDimensions()	514
10.140.3.8	SetOutsideValuePixel()	514
10.140.3.9	SetPixelMinMax()	515
10.140.3.10	SetPixmap()	515
10.141	dcm::ignore_char Struct Reference	515
10.141.1	Constructor & Destructor Documentation	515
10.141.1.1	ignore_char()	516

10.141.2	Member Data Documentation . . . . .	516
10.141.2.1	m_char . . . . .	516
10.142	dcmm::Image Class Reference . . . . .	516
10.142.1	Detailed Description . . . . .	518
10.142.2	Constructor & Destructor Documentation . . . . .	518
10.142.2.1	Image() . . . . .	518
10.142.2.2	~Image() . . . . .	519
10.142.3	Member Function Documentation . . . . .	519
10.142.3.1	GetDirectionCosines() [1/2] . . . . .	519
10.142.3.2	GetDirectionCosines() [2/2] . . . . .	519
10.142.3.3	GetIntercept() . . . . .	519
10.142.3.4	GetOrigin() [1/2] . . . . .	519
10.142.3.5	GetOrigin() [2/2] . . . . .	520
10.142.3.6	GetSlope() . . . . .	520
10.142.3.7	GetSpacing() [1/2] . . . . .	520
10.142.3.8	GetSpacing() [2/2] . . . . .	520
10.142.3.9	Print() . . . . .	520
10.142.3.10	SetDirectionCosines() [1/3] . . . . .	521
10.142.3.11	SetDirectionCosines() [2/3] . . . . .	521
10.142.3.12	SetDirectionCosines() [3/3] . . . . .	521
10.142.3.13	SetIntercept() . . . . .	521
10.142.3.14	SetOrigin() [1/3] . . . . .	521
10.142.3.15	SetOrigin() [2/3] . . . . .	522
10.142.3.16	SetOrigin() [3/3] . . . . .	522
10.142.3.17	SetSlope() . . . . .	522
10.142.3.18	SetSpacing() [1/2] . . . . .	522
10.142.3.19	SetSpacing() [2/2] . . . . .	522
10.143	dcmm::ImageApplyLookupTable Class Reference . . . . .	523

10.143.1	Detailed Description	525
10.143.2	Constructor & Destructor Documentation	525
10.143.2.1	ImageApplyLookupTable()	525
10.143.2.2	~ImageApplyLookupTable()	525
10.143.3	Member Function Documentation	525
10.143.3.1	Apply()	525
10.144	dcM::ImageChangePhotometricInterpretation Class Reference	526
10.144.1	Detailed Description	528
10.144.2	Constructor & Destructor Documentation	528
10.144.2.1	ImageChangePhotometricInterpretation()	528
10.144.2.2	~ImageChangePhotometricInterpretation()	528
10.144.3	Member Function Documentation	528
10.144.3.1	Change()	529
10.144.3.2	ChangeMonochrome()	529
10.144.3.3	ChangeRGB2YBR()	529
10.144.3.4	ChangeYBR2RGB()	529
10.144.3.5	GetPhotometricInterpretation()	529
10.144.3.6	RGB2YBR()	529
10.144.3.7	SetPhotometricInterpretation()	530
10.144.3.8	YBR2RGB()	530
10.145	dcM::ImageChangePlanarConfiguration Class Reference	530
10.145.1	Detailed Description	532
10.145.2	Constructor & Destructor Documentation	532
10.145.2.1	ImageChangePlanarConfiguration()	532
10.145.2.2	~ImageChangePlanarConfiguration()	532
10.145.3	Member Function Documentation	532
10.145.3.1	Change()	532
10.145.3.2	GetPlanarConfiguration()	533

10.145.3.3	<a href="#">RGBPixelsToRGBPlanes()</a> . . . . .	533
10.145.3.4	<a href="#">RGBPlanesToRGBPixels()</a> . . . . .	533
10.145.3.5	<a href="#">SetPlanarConfiguration()</a> . . . . .	533
10.146	<a href="#">dcm::ImageChangeTransferSyntax Class Reference</a> . . . . .	534
10.146.1	<a href="#">Detailed Description</a> . . . . .	536
10.146.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	536
10.146.2.1	<a href="#">ImageChangeTransferSyntax()</a> . . . . .	536
10.146.2.2	<a href="#">~ImageChangeTransferSyntax()</a> . . . . .	536
10.146.3	<a href="#">Member Function Documentation</a> . . . . .	537
10.146.3.1	<a href="#">Change()</a> . . . . .	537
10.146.3.2	<a href="#">GetTransferSyntax()</a> . . . . .	537
10.146.3.3	<a href="#">SetCompressIconImage()</a> . . . . .	537
10.146.3.4	<a href="#">SetForce()</a> . . . . .	537
10.146.3.5	<a href="#">SetTransferSyntax()</a> . . . . .	538
10.146.3.6	<a href="#">SetUserCodec()</a> . . . . .	538
10.146.3.7	<a href="#">TryJPEG2000Codec()</a> . . . . .	538
10.146.3.8	<a href="#">TryJPEGCodec()</a> . . . . .	538
10.146.3.9	<a href="#">TryJPEGLSCodec()</a> . . . . .	539
10.146.3.10	<a href="#">TryRAWCodec()</a> . . . . .	539
10.146.3.11	<a href="#">TryRLECodec()</a> . . . . .	539
10.147	<a href="#">dcm::ImageCodec Class Reference</a> . . . . .	539
10.147.1	<a href="#">Detailed Description</a> . . . . .	541
10.147.2	<a href="#">Member Typedef Documentation</a> . . . . .	542
10.147.2.1	<a href="#">LUTPtr</a> . . . . .	542
10.147.3	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	542
10.147.3.1	<a href="#">ImageCodec()</a> . . . . .	542
10.147.3.2	<a href="#">~ImageCodec()</a> . . . . .	542
10.147.4	<a href="#">Member Function Documentation</a> . . . . .	542

10.147.4.1AppendFrameEncode()	542
10.147.4.2AppendRowEncode()	543
10.147.4.3CanCode()	543
10.147.4.4CanDecode()	543
10.147.4.5CleanupUnusedBits()	543
10.147.4.6Clone()	544
10.147.4.7Decode()	544
10.147.4.8DecodeByStreams()	544
10.147.4.9DoByteSwap()	544
10.147.4.10DoInvertMonochrome()	545
10.147.4.11DoOverlayCleanup()	545
10.147.4.12DoPaddedCompositePixelCode()	545
10.147.4.13DoPlanarConfiguration()	545
10.147.4.14DoSimpleCopy()	545
10.147.4.15DoYBR()	545
10.147.4.16GetDimensions()	546
10.147.4.17GetHeaderInfo()	546
10.147.4.18GetLossyFlag()	546
10.147.4.19GetLUT()	546
10.147.4.20GetNeedByteSwap()	546
10.147.4.21GetNumberOfDimensions()	546
10.147.4.22GetPhotometricInterpretation()	547
10.147.4.23GetPixelFormat() [1/2]	547
10.147.4.24GetPixelFormat() [2/2]	547
10.147.4.25GetPlanarConfiguration()	547
10.147.4.26FrameEncoder()	547
10.147.4.27Lossy()	547
10.147.4.28RowEncoder()	548

10.147.4.29	<a href="#">Valid()</a>	548
10.147.4.30	<a href="#">SetDimensions() [1/2]</a>	548
10.147.4.31	<a href="#">SetDimensions() [2/2]</a>	548
10.147.4.32	<a href="#">SetLossyFlag()</a>	548
10.147.4.33	<a href="#">SetLUT()</a>	549
10.147.4.34	<a href="#">SetNeedByteSwap()</a>	549
10.147.4.35	<a href="#">SetNeedOverlayCleanup()</a>	549
10.147.4.36	<a href="#">SetNumberOfDimensions()</a>	549
10.147.4.37	<a href="#">SetPhotometricInterpretation()</a>	549
10.147.4.38	<a href="#">SetPixelFormat()</a>	550
10.147.4.39	<a href="#">SetPlanarConfiguration()</a>	550
10.147.4.40	<a href="#">StartEncode()</a>	550
10.147.4.41	<a href="#">StopEncode()</a>	550
10.147.5	<a href="#">Friends And Related Function Documentation</a>	550
10.147.5.1	<a href="#">FileChangeTransferSyntax</a>	551
10.147.5.2	<a href="#">ImageChangePhotometricInterpretation</a>	551
10.147.6	<a href="#">Member Data Documentation</a>	551
10.147.6.1	<a href="#">Dimensions</a>	551
10.147.6.2	<a href="#">LossyFlag</a>	551
10.147.6.3	<a href="#">LUT</a>	551
10.147.6.4	<a href="#">NeedByteSwap</a>	551
10.147.6.5	<a href="#">NeedOverlayCleanup</a>	552
10.147.6.6	<a href="#">NumberOfDimensions</a>	552
10.147.6.7	<a href="#">PF</a>	552
10.147.6.8	<a href="#">PI</a>	552
10.147.6.9	<a href="#">PlanarConfiguration</a>	552
10.147.6.10	<a href="#">RequestPaddedCompositePixelCode</a>	552
10.147.6.11	<a href="#">RequestPlanarConfiguration</a>	552

10.148	dcm::ImageConverter Class Reference	553
10.148.1	Detailed Description	553
10.148.2	Constructor & Destructor Documentation	553
10.148.2.1	ImageConverter()	553
10.148.2.2	~ImageConverter()	553
10.148.3	Member Function Documentation	553
10.148.3.1	Convert()	554
10.148.3.2	GetOutput()	554
10.148.3.3	SetInput()	554
10.149	dcm::ImageFragmentSplitter Class Reference	554
10.149.1	Detailed Description	556
10.149.2	Constructor & Destructor Documentation	556
10.149.2.1	ImageFragmentSplitter()	556
10.149.2.2	~ImageFragmentSplitter()	556
10.149.3	Member Function Documentation	556
10.149.3.1	GetFragmentSizeMax()	556
10.149.3.2	SetForce()	556
10.149.3.3	SetFragmentSizeMax()	557
10.149.3.4	Split()	557
10.150	dcm::ImageHelper Class Reference	557
10.150.1	Detailed Description	558
10.150.2	Member Function Documentation	558
10.150.2.1	ComputeMediaStorageFromModality()	558
10.150.2.2	ComputeSpacingFromImagePositionPatient()	559
10.150.2.3	GetDimensionsValue()	559
10.150.2.4	GetDirectionCosinesFromDataSet()	559
10.150.2.5	GetDirectionCosinesValue()	559
10.150.2.6	GetForcePixelSpacing()	559

10.150.2.7	GetForceRescaleInterceptSlope()	560
10.150.2.8	GetLUT()	560
10.150.2.9	GetOriginValue()	560
10.150.2.10	GetPhotometricInterpretationValue()	560
10.150.2.11	GetPixelFormatValue()	560
10.150.2.12	GetPlanarConfigurationValue()	560
10.150.2.13	GetPMSRescaleInterceptSlope()	561
10.150.2.14	GetPointerFromElement()	561
10.150.2.15	GetRealWorldValueMappingContent()	561
10.150.2.16	GetRescaleInterceptSlopeValue()	561
10.150.2.17	GetSpacingTagFromMediaStorage()	561
10.150.2.18	GetSpacingValue()	562
10.150.2.19	GetZSpacingTagFromMediaStorage()	562
10.150.2.20	GetDimensionsValue()	562
10.150.2.21	GetDirectionCosinesValue()	562
10.150.2.22	GetForcePixelSpacing()	562
10.150.2.23	GetForceRescaleInterceptSlope()	563
10.150.2.24	GetOriginValue()	563
10.150.2.25	GetPMSRescaleInterceptSlope()	563
10.150.2.26	GetRescaleInterceptSlopeValue()	563
10.150.2.27	GetSpacingValue()	563
10.151	dcm::ImageReader Class Reference	564
10.151.1	Detailed Description	566
10.151.2	Constructor & Destructor Documentation	566
10.151.2.1	ImageReader()	566
10.151.2.2	~ImageReader()	566
10.151.3	Member Function Documentation	566
10.151.3.1	GetImage() [1/2]	567



10.151.3.2	<a href="#">GetImage()</a> [2/2]	567
10.151.3.3	<a href="#">Read()</a>	567
10.151.3.4	<a href="#">ReadACRNEMAImage()</a>	567
10.151.3.5	<a href="#">ReadImage()</a>	568
10.152	<a href="#">dcm::ImageRegionReader Class Reference</a>	568
10.152.1	<a href="#">Detailed Description</a>	570
10.152.2	<a href="#">Constructor &amp; Destructor Documentation</a>	570
10.152.2.1	<a href="#">ImageRegionReader()</a>	570
10.152.2.2	<a href="#">~ImageRegionReader()</a>	570
10.152.3	<a href="#">Member Function Documentation</a>	570
10.152.3.1	<a href="#">ComputeBufferLength()</a>	571
10.152.3.2	<a href="#">GetRegion()</a>	571
10.152.3.3	<a href="#">Read()</a>	571
10.152.3.4	<a href="#">ReadInformation()</a>	571
10.152.3.5	<a href="#">ReadIntoBuffer()</a>	572
10.152.3.6	<a href="#">SetRegion()</a>	572
10.153	<a href="#">dcm::ImageToImageFilter Class Reference</a>	572
10.153.1	<a href="#">Detailed Description</a>	573
10.153.2	<a href="#">Constructor &amp; Destructor Documentation</a>	574
10.153.2.1	<a href="#">ImageToImageFilter()</a>	574
10.153.2.2	<a href="#">~ImageToImageFilter()</a>	574
10.153.3	<a href="#">Member Function Documentation</a>	574
10.153.3.1	<a href="#">GetInput()</a>	574
10.153.3.2	<a href="#">GetOutput()</a>	574
10.154	<a href="#">dcm::ImageWriter Class Reference</a>	575
10.154.1	<a href="#">Detailed Description</a>	577
10.154.2	<a href="#">Constructor &amp; Destructor Documentation</a>	577
10.154.2.1	<a href="#">ImageWriter()</a>	577

10.154.2.2~ImageWriter()	577
10.154.3Member Function Documentation	577
10.154.3.1ComputeTargetMediaStorage()	577
10.154.3.2GetImage() [1/2]	578
10.154.3.3GetImage() [2/2]	578
10.154.3.4Write()	578
10.155dcm::network::ImplementationClassUIDSub Class Reference	578
10.155.1Detailed Description	579
10.155.2Constructor & Destructor Documentation	579
10.155.2.1ImplementationClassUIDSub()	579
10.155.3Member Function Documentation	579
10.155.3.1Print()	579
10.155.3.2Read()	579
10.155.3.3Size()	580
10.155.3.4Write()	580
10.156dcm::network::ImplementationUIDSub Class Reference	580
10.156.1Detailed Description	580
10.156.2Constructor & Destructor Documentation	580
10.156.2.1ImplementationUIDSub()	580
10.156.3Member Function Documentation	581
10.156.3.1Write()	581
10.157dcm::network::ImplementationVersionNameSub Class Reference	581
10.157.1Detailed Description	581
10.157.2Constructor & Destructor Documentation	581
10.157.2.1ImplementationVersionNameSub()	581
10.157.3Member Function Documentation	582
10.157.3.1Print()	582
10.157.3.2Read()	582

10.157.3.3	Size()	582
10.157.3.4	Write()	582
10.158	dcm::ImplicitDataElement Class Reference	583
10.158.1	Detailed Description	584
10.158.2	Member Function Documentation	584
10.158.2.1	GetLength()	584
10.158.2.2	Read()	584
10.158.2.3	ReadPreValue()	585
10.158.2.4	ReadValue()	585
10.158.2.5	ReadValueWithLength()	585
10.158.2.6	ReadWithLength()	585
10.158.2.7	Write()	585
10.159	dcm::InitializeEvent Class Reference	586
10.160	dcm::IOD Class Reference	587
10.160.1	Detailed Description	587
10.160.2	Member Typedef Documentation	587
10.160.2.1	MapIODEntry	588
10.160.2.2	SizeType	588
10.160.3	Constructor & Destructor Documentation	588
10.160.3.1	IOD()	588
10.160.4	Member Function Documentation	588
10.160.4.1	AddIODEntry()	588
10.160.4.2	Clear()	588
10.160.4.3	GetIODEntry()	589
10.160.4.4	GetNumberOfIODs()	589
10.160.4.5	GetTypeFromTag()	589
10.160.5	Friends And Related Function Documentation	589
10.160.5.1	operator<<	589

10.161.0 dcm::IODEntry Class Reference . . . . .	590
10.161.1 Detailed Description . . . . .	590
10.161.2 Constructor & Destructor Documentation . . . . .	591
10.161.2.1 IODEntry() . . . . .	591
10.161.3 Member Function Documentation . . . . .	591
10.161.3.1 GetIE() . . . . .	591
10.161.3.2 GetName() . . . . .	591
10.161.3.3 GetRef() . . . . .	591
10.161.3.4 GetUsage() . . . . .	591
10.161.3.5 GetUsageType() . . . . .	592
10.161.3.6 SetIE() . . . . .	592
10.161.3.7 SetName() . . . . .	592
10.161.3.8 SetRef() . . . . .	592
10.161.3.9 SetUsage() . . . . .	592
10.161.4 Friends And Related Function Documentation . . . . .	592
10.161.4.1 operator<< . . . . .	592
10.162.0 dcm::IODs Class Reference . . . . .	593
10.162.1 Detailed Description . . . . .	593
10.162.2 Member Typedef Documentation . . . . .	593
10.162.2.1 IODMapType . . . . .	594
10.162.2.2 IODMapTypeConstIterator . . . . .	594
10.162.2.3 IODName . . . . .	594
10.162.3 Constructor & Destructor Documentation . . . . .	594
10.162.3.1 IODs() . . . . .	594
10.162.4 Member Function Documentation . . . . .	594
10.162.4.1 AddIOD() . . . . .	594
10.162.4.2 Begin() . . . . .	595
10.162.4.3 Clear() . . . . .	595

10.162.4.4	End()	595
10.162.4.5	GetIOD()	595
10.162.5	Friends And Related Function Documentation	595
10.162.5.1	operator<<	595
10.163	dcm::IPPSorter Class Reference	596
10.163.1	Detailed Description	597
10.163.2	Constructor & Destructor Documentation	597
10.163.2.1	IPPSorter()	597
10.163.3	Member Function Documentation	598
10.163.3.1	GetDirectionCosinesTolerance()	598
10.163.3.2	GetZSpacing()	598
10.163.3.3	GetZSpacingTolerance()	598
10.163.3.4	SetComputeZSpacing()	598
10.163.3.5	SetDirectionCosinesTolerance()	599
10.163.3.6	SetDropDuplicatePositions()	599
10.163.3.7	SetZSpacingTolerance()	599
10.163.3.8	Sort()	599
10.163.4	Member Data Documentation	600
10.163.4.1	ComputeZSpacing	600
10.163.4.2	DirCosTolerance	600
10.163.4.3	DropDuplicatePositions	600
10.163.4.4	ZSpacing	600
10.163.4.5	ZTolerance	600
10.164	dcm::Item Class Reference	601
10.164.1	Detailed Description	602
10.164.2	Constructor & Destructor Documentation	602
10.164.2.1	Item() [1/2]	603
10.164.2.2	Item() [2/2]	603

10.164.3	Member Function Documentation	603
10.164.3.1	Clear()	603
10.164.3.2	FindDataElement()	603
10.164.3.3	GetDataElement()	603
10.164.3.4	GetLength()	604
10.164.3.5	GetNestedDataSet() [1/2]	604
10.164.3.6	GetNestedDataSet() [2/2]	604
10.164.3.7	InsertDataElement()	604
10.164.3.8	Read()	604
10.164.3.9	SetNestedDataSet()	605
10.164.3.10	Write()	605
10.164.4	Friends And Related Function Documentation	605
10.164.4.1	operator<<	605
10.165	dcm::IterationEvent Class Reference	605
10.166	dcm::JPEG12Codec Class Reference	606
10.166.1	Detailed Description	608
10.166.2	Constructor & Destructor Documentation	608
10.166.2.1	JPEG12Codec()	608
10.166.2.2	~JPEG12Codec()	608
10.166.3	Member Function Documentation	608
10.166.3.1	DecodeByStreams()	608
10.166.3.2	EncodeBuffer()	609
10.166.3.3	GetHeaderInfo()	609
10.166.3.4	InternalCode()	609
10.166.3.5	IsStateSuspension()	609
10.167	dcm::JPEG16Codec Class Reference	610
10.167.1	Detailed Description	611
10.167.2	Constructor & Destructor Documentation	611

10.167.2.1Jpeg16Codec()	611
10.167.2.2~Jpeg16Codec()	611
10.167.3Member Function Documentation	611
10.167.3.1DecodeByStreams()	612
10.167.3.2EncodeBuffer()	612
10.167.3.3GetHeaderInfo()	612
10.167.3.4InternalCode()	612
10.167.3.5IsStateSuspension()	612
10.168Jcdm::Jpeg2000Codec Class Reference	613
10.168.1Detailed Description	614
10.168.2Constructor & Destructor Documentation	615
10.168.2.1Jpeg2000Codec()	615
10.168.2.2~Jpeg2000Codec()	615
10.168.3Member Function Documentation	615
10.168.3.1AppendFrameEncode()	615
10.168.3.2AppendRowEncode()	615
10.168.3.3CanCode()	616
10.168.3.4CanDecode()	616
10.168.3.5Clone()	616
10.168.3.6Code()	616
10.168.3.7Decode()	617
10.168.3.8DecodeByStreams()	617
10.168.3.9DecodeExtent()	617
10.168.3.10GetHeaderInfo()	617
10.168.3.11GetQuality()	618
10.168.3.12GetRate()	618
10.168.3.13FrameEncoder()	618
10.168.3.14RowEncoder()	618

10.168.3.1	<del>Set</del> NumberOfResolutions()	618
10.168.3.1	<del>Set</del> Quality()	618
10.168.3.1	<del>Set</del> Rate()	619
10.168.3.1	<del>Set</del> Reversible()	619
10.168.3.1	<del>Set</del> TileSize()	619
10.168.3.2	<del>Start</del> Encode()	619
10.168.3.2	<del>Stop</del> Encode()	619
10.168.4	Friends And Related Function Documentation	619
10.168.4.1	Bitmap	620
10.168.4.2	ImageRegionReader	620
10.169	dcm::JPEG8Codec Class Reference	620
10.169.1	Detailed Description	621
10.169.2	Constructor & Destructor Documentation	621
10.169.2.1	JPEG8Codec()	621
10.169.2.2	~JPEG8Codec()	622
10.169.3	Member Function Documentation	622
10.169.3.1	DecodeByStreams()	622
10.169.3.2	EncodeBuffer()	622
10.169.3.3	GetHeaderInfo()	622
10.169.3.4	InternalCode()	622
10.169.3.5	IsStateSuspension()	623
10.170	dcm::JPEGCodec Class Reference	623
10.170.1	Detailed Description	625
10.170.2	Constructor & Destructor Documentation	625
10.170.2.1	JPEGCodec()	625
10.170.2.2	~JPEGCodec()	625
10.170.3	Member Function Documentation	625
10.170.3.1	AppendFrameEncode()	626



10.170.3.2AppendRowEncode()	626
10.170.3.3CanCode()	626
10.170.3.4CanDecode()	626
10.170.3.5Clone()	627
10.170.3.6Code()	627
10.170.3.7ComputeOffsetTable()	627
10.170.3.8Decode()	627
10.170.3.9DecodeByStreams()	627
10.170.3.10DecodeExtent()	628
10.170.3.11EncodeBuffer()	628
10.170.3.12GetHeaderInfo()	628
10.170.3.13GetLossless()	628
10.170.3.14GetQuality()	629
10.170.3.15FrameEncoder()	629
10.170.3.16RowEncoder()	629
10.170.3.17StateSuspension()	629
10.170.3.18Valid()	629
10.170.3.19SetBitSample()	629
10.170.3.20SetLossless()	630
10.170.3.21SetPixelFormat()	630
10.170.3.22SetQuality()	630
10.170.3.23StartEncode()	630
10.170.3.24StopEncode()	630
10.170.4Friends And Related Function Documentation	631
10.170.4.1ImageRegionReader	631
10.170.5Member Data Documentation	631
10.170.5.1BitSample	631
10.170.5.2Quality	631

10.171.0 dcm::JPEGLSCodec Class Reference . . . . .	631
10.171.1 Detailed Description . . . . .	633
10.171.2 Constructor & Destructor Documentation . . . . .	633
10.171.2.1 JPEGLSCodec() . . . . .	634
10.171.2.2 ~JPEGLSCodec() . . . . .	634
10.171.3 Member Function Documentation . . . . .	634
10.171.3.1 AppendFrameEncode() . . . . .	634
10.171.3.2 AppendRowEncode() . . . . .	634
10.171.3.3 CanCode() . . . . .	634
10.171.3.4 CanDecode() . . . . .	635
10.171.3.5 Clone() . . . . .	635
10.171.3.6 Code() . . . . .	635
10.171.3.7 Decode() [1/2] . . . . .	635
10.171.3.8 Decode() [2/2] . . . . .	636
10.171.3.9 DecodeExtent() . . . . .	636
10.171.3.10 GetBufferLength() . . . . .	636
10.171.3.11 GetHeaderInfo() . . . . .	636
10.171.3.12 GetLossless() . . . . .	637
10.171.3.13 FrameEncoder() . . . . .	637
10.171.3.14 RowEncoder() . . . . .	637
10.171.3.15 SetBufferLength() . . . . .	637
10.171.3.16 SetLossless() . . . . .	637
10.171.3.17 SetLossyError() . . . . .	637
10.171.3.18 StartEncode() . . . . .	638
10.171.3.19 StopEncode() . . . . .	638
10.171.4 Friends And Related Function Documentation . . . . .	638
10.171.4.1 ImageRegionReader . . . . .	638
10.171.0 dcm::JSON Class Reference . . . . .	638

10.172.1Detailed Description . . . . .	639
10.172.2Constructor & Destructor Documentation . . . . .	639
10.172.2.1JSON() . . . . .	639
10.172.2.2~JSON() . . . . .	639
10.172.3Member Function Documentation . . . . .	639
10.172.3.1Code() . . . . .	639
10.172.3.2Decode() . . . . .	639
10.172.3.3GetPrettyPrint() . . . . .	640
10.172.3.4PrettyPrintOff() . . . . .	640
10.172.3.5PrettyPrintOn() . . . . .	640
10.172.3.6SetPrettyPrint() . . . . .	640
10.173dcm::KAKADUCodec Class Reference . . . . .	641
10.173.1Detailed Description . . . . .	642
10.173.2Constructor & Destructor Documentation . . . . .	642
10.173.2.1KAKADUCodec() . . . . .	642
10.173.2.2~KAKADUCodec() . . . . .	642
10.173.3Member Function Documentation . . . . .	642
10.173.3.1CanCode() . . . . .	642
10.173.3.2CanDecode() . . . . .	643
10.173.3.3Clone() . . . . .	643
10.173.3.4Code() . . . . .	643
10.173.3.5Decode() . . . . .	643
10.174dcm::LO Class Reference . . . . .	644
10.174.1Detailed Description . . . . .	645
10.174.2Member Typedef Documentation . . . . .	645
10.174.2.1const_iterator . . . . .	645
10.174.2.2const_reference . . . . .	645
10.174.2.3const_reverse_iterator . . . . .	645

10.174.2.4	difference_type	645
10.174.2.5	iterator	646
10.174.2.6	pointer	646
10.174.2.7	reference	646
10.174.2.8	reverse_iterator	646
10.174.2.9	size_type	646
10.174.2.10	superclass	646
10.174.2.11	value_type	646
10.174.3	Constructor & Destructor Documentation	647
10.174.3.1	LO() [1/4]	647
10.174.3.2	LO() [2/4]	647
10.174.3.3	LO() [3/4]	647
10.174.3.4	LO() [4/4]	647
10.174.4	Member Function Documentation	647
10.174.4.1	IsValid()	647
10.175	dcm::LookupTable Class Reference	648
10.175.1	Detailed Description	649
10.175.2	Member Enumeration Documentation	650
10.175.2.1	LookupTableType	650
10.175.3	Constructor & Destructor Documentation	650
10.175.3.1	LookupTable() [1/2]	650
10.175.3.2	~LookupTable()	650
10.175.3.3	LookupTable() [2/2]	650
10.175.4	Member Function Documentation	650
10.175.4.1	Allocate()	651
10.175.4.2	Clear()	651
10.175.4.3	Decode() [1/2]	651
10.175.4.4	Decode() [2/2]	651

10.175.4.5GetBitSample()	651
10.175.4.6GetBufferAsRGBA()	652
10.175.4.7GetLUT()	652
10.175.4.8GetLUTDescriptor()	652
10.175.4.9GetLUTLength()	652
10.175.4.10GetPointer()	652
10.175.4.11InitializeBlueLUT()	653
10.175.4.12Initialized()	653
10.175.4.13InitializeGreenLUT()	653
10.175.4.14InitializeLUT()	653
10.175.4.15InitializeRedLUT()	653
10.175.4.16Print()	654
10.175.4.17SetBlueLUT()	654
10.175.4.18SetGreenLUT()	654
10.175.4.19SetLUT()	654
10.175.4.20SetRedLUT()	654
10.175.4.21WriteBufferAsRGBA()	655
10.175.5Member Data Documentation	655
10.175.5.1BitSample	655
10.175.5.2IncompleteLUT	655
10.175.5.3Internal	655
10.176dcm::Scanner::Itstr Struct Reference	655
10.176.1Member Function Documentation	656
10.176.1.1operator()()	656
10.177dcm::StrictScanner::Itstr Struct Reference	656
10.177.1Member Function Documentation	656
10.177.1.1operator()()	656
10.178dcm::Macro Class Reference	656

10.178.1Detailed Description . . . . .	657
10.178.2Member Typedef Documentation . . . . .	657
10.178.2.1ArrayIncludeMacrosType . . . . .	657
10.178.2.2MapModuleEntry . . . . .	658
10.178.3Constructor & Destructor Documentation . . . . .	658
10.178.3.1Macro() . . . . .	658
10.178.4Member Function Documentation . . . . .	658
10.178.4.1AddMacroEntry() . . . . .	658
10.178.4.2Clear() . . . . .	658
10.178.4.3FindMacroEntry() . . . . .	658
10.178.4.4GetMacroEntry() . . . . .	659
10.178.4.5GetName() . . . . .	659
10.178.4.6SetName() . . . . .	659
10.178.4.7Verify() . . . . .	659
10.178.5Friends And Related Function Documentation . . . . .	659
10.178.5.1operator<< . . . . .	659
10.179dcm::Macros Class Reference . . . . .	660
10.179.1Detailed Description . . . . .	660
10.179.2Member Typedef Documentation . . . . .	660
10.179.2.1ModuleMapType . . . . .	661
10.179.3Constructor & Destructor Documentation . . . . .	661
10.179.3.1Macros() . . . . .	661
10.179.4Member Function Documentation . . . . .	661
10.179.4.1AddMacro() . . . . .	661
10.179.4.2Clear() . . . . .	661
10.179.4.3GetMacro() . . . . .	661
10.179.4.4IsEmpty() . . . . .	661
10.179.5Friends And Related Function Documentation . . . . .	662

10.179.5.1operator<<	662
10.180dcm::network::MaximumLengthSub Class Reference	662
10.180.1Detailed Description	662
10.180.2Constructor & Destructor Documentation	662
10.180.2.1MaximumLengthSub()	663
10.180.3Member Function Documentation	663
10.180.3.1GetMaximumLength()	663
10.180.3.2Print()	663
10.180.3.3Read()	663
10.180.3.4SetMaximumLength()	663
10.180.3.5Size()	664
10.180.3.6Write()	664
10.181dcm::MD5 Class Reference	664
10.181.1Detailed Description	664
10.181.2Member Function Documentation	665
10.181.2.1Compute()	665
10.181.2.2ComputeFile()	665
10.182dcm::MediaStorage Class Reference	665
10.182.1Detailed Description	668
10.182.2Member Enumeration Documentation	668
10.182.2.1MSType	668
10.182.2.2ObjectType	671
10.182.3Constructor & Destructor Documentation	671
10.182.3.1MediaStorage()	671
10.182.4Member Function Documentation	671
10.182.4.1GetModality()	672
10.182.4.2GetModalityDimension()	672
10.182.4.3GetMSString()	672

10.182.4.4	GetMSType()	672
10.182.4.5	GetNumberOfModality()	672
10.182.4.6	GetNumberOfMSString()	673
10.182.4.7	GetNumberOfMSType()	673
10.182.4.8	GetString()	673
10.182.4.9	GuessFromModality()	673
10.182.4.10	Image()	673
10.182.4.11	Undefined()	674
10.182.4.12	operator MSType()	674
10.182.4.13	SetFromDataSet()	674
10.182.4.13	SetFromFile()	674
10.182.4.15	SetFromHeader()	674
10.182.4.16	SetFromModality()	675
10.182.4.17	SetFromSourceImageSequence()	675
10.182.5	Friends And Related Function Documentation	675
10.182.5.1	operator<<	675
10.183	dcm::MemberCommand< T > Class Template Reference	675
10.183.1	Detailed Description	677
10.183.2	Member Typedef Documentation	677
10.183.2.1	Self	677
10.183.2.2	TConstMemberFunctionPointer	678
10.183.2.3	TMemberFunctionPointer	678
10.183.3	Constructor & Destructor Documentation	678
10.183.3.1	MemberCommand()	678
10.183.3.2	~MemberCommand()	678
10.183.4	Member Function Documentation	678
10.183.4.1	Execute() [1/2]	678
10.183.4.2	Execute() [2/2]	679



10.183.4.3New()	679
10.183.4.4SetCallbackFunction() [1/2]	679
10.183.4.5SetCallbackFunction() [2/2]	679
10.183.5Member Data Documentation	679
10.183.5.1m_ConstMemberFunction	679
10.183.5.2m_MemberFunction	680
10.183.5.3m_This	680
10.184dcm::MeshPrimitive Class Reference	680
10.184.1Detailed Description	682
10.184.2Member Typedef Documentation	682
10.184.2.1PrimitivesData	682
10.184.3Member Enumeration Documentation	682
10.184.3.1MPType	682
10.184.4Constructor & Destructor Documentation	683
10.184.4.1MeshPrimitive()	683
10.184.4.2~MeshPrimitive()	683
10.184.5Member Function Documentation	683
10.184.5.1AddPrimitiveData()	683
10.184.5.2GetMPType()	683
10.184.5.3GetMPTypeString()	684
10.184.5.4GetNumberOfPrimitivesData()	684
10.184.5.5GetPrimitiveData() [1/4]	684
10.184.5.6GetPrimitiveData() [2/4]	684
10.184.5.7GetPrimitiveData() [3/4]	684
10.184.5.8GetPrimitiveData() [4/4]	684
10.184.5.9GetPrimitivesData() [1/2]	684
10.184.5.10GetPrimitivesData() [2/2]	685
10.184.5.11GetPrimitiveType()	685

10.184.5.1SetPrimitiveData() [1/2]	685
10.184.5.1SetPrimitiveData() [2/2]	685
10.184.5.1SetPrimitivesData()	685
10.184.5.1SetPrimitiveType()	685
10.184.6Member Data Documentation	685
10.184.6.1PrimitiveData	686
10.184.6.2PrimitiveType	686
10.185dcm::ModalityPerformedProcedureStepCreateQuery Class Reference	686
10.185.1Detailed Description	687
10.185.2Constructor & Destructor Documentation	688
10.185.2.1ModalityPerformedProcedureStepCreateQuery()	688
10.185.3Member Function Documentation	688
10.185.3.1GetAbstractSyntaxUID()	688
10.185.3.2GetRequiredDataSet()	688
10.185.3.3ValidateQuery()	688
10.185.4Friends And Related Function Documentation	688
10.185.4.1QueryFactory	689
10.186dcm::ModalityPerformedProcedureStepSetQuery Class Reference	689
10.186.1Detailed Description	690
10.186.2Constructor & Destructor Documentation	691
10.186.2.1ModalityPerformedProcedureStepSetQuery()	691
10.186.3Member Function Documentation	691
10.186.3.1GetAbstractSyntaxUID()	691
10.186.3.2GetRequiredDataSet()	691
10.186.3.3ValidateQuery()	691
10.186.4Friends And Related Function Documentation	691
10.186.4.1QueryFactory	692
10.187dcm::ModifiedEvent Class Reference	692

10.188	dcm::Module Class Reference . . . . .	693
10.188.1	Detailed Description . . . . .	693
10.188.2	Member Typedef Documentation . . . . .	694
10.188.2.1	ArrayIncludeMacrosType . . . . .	694
10.188.2.2	MapModuleEntry . . . . .	694
10.188.3	Constructor & Destructor Documentation . . . . .	694
10.188.3.1	Module() . . . . .	694
10.188.4	Member Function Documentation . . . . .	694
10.188.4.1	AddMacro() . . . . .	694
10.188.4.2	AddModuleEntry() . . . . .	694
10.188.4.3	Clear() . . . . .	695
10.188.4.4	FindModuleEntryInMacros() . . . . .	695
10.188.4.5	GetModuleEntryInMacros() . . . . .	695
10.188.4.6	GetName() . . . . .	695
10.188.4.7	SetName() . . . . .	695
10.188.4.8	Verify() . . . . .	696
10.188.5	Friends And Related Function Documentation . . . . .	696
10.188.5.1	operator<< . . . . .	696
10.189	dcm::ModuleEntry Class Reference . . . . .	696
10.189.1	Detailed Description . . . . .	698
10.189.2	Member Typedef Documentation . . . . .	698
10.189.2.1	Description . . . . .	698
10.189.3	Constructor & Destructor Documentation . . . . .	698
10.189.3.1	ModuleEntry() . . . . .	698
10.189.3.2	~ModuleEntry() . . . . .	698
10.189.4	Member Function Documentation . . . . .	699
10.189.4.1	GetDescription() . . . . .	699
10.189.4.2	GetName() . . . . .	699

10.189.4.3	<a href="#">GetType()</a> . . . . .	699
10.189.4.4	<a href="#">SetDescription()</a> . . . . .	699
10.189.4.5	<a href="#">SetName()</a> . . . . .	699
10.189.4.6	<a href="#">SetType()</a> . . . . .	699
10.189.5	<a href="#">Friends And Related Function Documentation</a> . . . . .	700
10.189.5.1	<a href="#">operator&lt;&lt;</a> . . . . .	700
10.189.6	<a href="#">Member Data Documentation</a> . . . . .	700
10.189.6.1	<a href="#">DataElementType</a> . . . . .	700
10.189.6.2	<a href="#">DescriptionField</a> . . . . .	700
10.189.6.3	<a href="#">Name</a> . . . . .	700
10.190	<a href="#">dcm::Modules Class Reference</a> . . . . .	701
10.190.1	<a href="#">Detailed Description</a> . . . . .	701
10.190.2	<a href="#">Member Typedef Documentation</a> . . . . .	701
10.190.2.1	<a href="#">ModuleMapType</a> . . . . .	702
10.190.3	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	702
10.190.3.1	<a href="#">Modules()</a> . . . . .	702
10.190.4	<a href="#">Member Function Documentation</a> . . . . .	702
10.190.4.1	<a href="#">AddModule()</a> . . . . .	702
10.190.4.2	<a href="#">Clear()</a> . . . . .	702
10.190.4.3	<a href="#">GetModule()</a> . . . . .	702
10.190.4.4	<a href="#">IsEmpty()</a> . . . . .	703
10.190.5	<a href="#">Friends And Related Function Documentation</a> . . . . .	703
10.190.5.1	<a href="#">operator&lt;&lt;</a> . . . . .	703
10.191	<a href="#">dcm::MovePatientRootQuery Class Reference</a> . . . . .	703
10.191.1	<a href="#">Detailed Description</a> . . . . .	704
10.191.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	704
10.191.2.1	<a href="#">MovePatientRootQuery()</a> . . . . .	704
10.191.3	<a href="#">Member Function Documentation</a> . . . . .	705

10.191.3.1	GetAbstractSyntaxUID()	705
10.191.3.2	GetTagListByLevel()	705
10.191.3.3	InitializeDataSet()	705
10.191.3.4	ValidateQuery()	705
10.191.4	Friends And Related Function Documentation	706
10.191.4.1	QueryFactory	706
10.192	dcm::MoveStudyRootQuery Class Reference	706
10.192.1	Detailed Description	707
10.192.2	Constructor & Destructor Documentation	707
10.192.2.1	MoveStudyRootQuery()	707
10.192.3	Member Function Documentation	708
10.192.3.1	GetAbstractSyntaxUID()	708
10.192.3.2	GetTagListByLevel()	708
10.192.3.3	InitializeDataSet()	708
10.192.3.4	ValidateQuery()	708
10.192.4	Friends And Related Function Documentation	709
10.192.4.1	QueryFactory	709
10.193	dcm::MrProtocol Class Reference	709
10.193.1	Detailed Description	710
10.193.2	Constructor & Destructor Documentation	710
10.193.2.1	MrProtocol()	710
10.193.2.2	~MrProtocol()	710
10.193.3	Member Function Documentation	710
10.193.3.1	FindMrProtocolByName()	710
10.193.3.2	GetMrProtocolByName()	710
10.193.3.3	GetSliceArray()	711
10.193.3.4	GetVersion()	711
10.193.3.5	Load()	711

10.193.3.6Print()	. 711
10.193.4Friends And Related Function Documentation	. 711
10.193.4.1operator<<	. 711
10.194dcm::network::NActionRQ Class Reference	. 712
10.194.1Detailed Description	. 712
10.194.2Member Function Documentation	. 713
10.194.2.1ConstructPDV()	. 713
10.195dcm::network::NActionRSP Class Reference	. 713
10.195.1Detailed Description	. 714
10.195.2Member Function Documentation	. 714
10.195.2.1ConstructPDVByDataSet()	. 714
10.196dcm::network::NCreateRQ Class Reference	. 715
10.196.1Detailed Description	. 715
10.196.2Member Function Documentation	. 716
10.196.2.1ConstructPDV()	. 716
10.197dcm::network::NCreateRSP Class Reference	. 716
10.197.1Detailed Description	. 717
10.197.2Member Function Documentation	. 717
10.197.2.1ConstructPDVByDataSet()	. 717
10.198dcm::network::NDeleteRQ Class Reference	. 718
10.198.1Detailed Description	. 718
10.198.2Member Function Documentation	. 719
10.198.2.1ConstructPDV()	. 719
10.199dcm::network::NDeleteRSP Class Reference	. 719
10.199.1Detailed Description	. 720
10.199.2Member Function Documentation	. 720
10.199.2.1ConstructPDVByDataSet()	. 720
10.200dcm::NestedModuleEntries Class Reference	. 721

10.200.1Detailed Description . . . . .	722
10.200.2Member Typedef Documentation . . . . .	722
10.200.2.1SizeType . . . . .	722
10.200.3Constructor & Destructor Documentation . . . . .	722
10.200.3.1NestedModuleEntries() . . . . .	723
10.200.4Member Function Documentation . . . . .	723
10.200.4.1AddModuleEntry() . . . . .	723
10.200.4.2GetModuleEntry() [1/2] . . . . .	723
10.200.4.3GetModuleEntry() [2/2] . . . . .	723
10.200.4.4GetNumberOfModuleEntries() . . . . .	723
10.200.5Friends And Related Function Documentation . . . . .	723
10.200.5.1operator<< . . . . .	724
10.201dcm::network::NEventReportRQ Class Reference . . . . .	724
10.201.1Detailed Description . . . . .	725
10.201.2Member Function Documentation . . . . .	725
10.201.2.1ConstructPDV() . . . . .	725
10.202dcm::network::NEventReportRSP Class Reference . . . . .	725
10.202.1Detailed Description . . . . .	726
10.202.2Member Function Documentation . . . . .	726
10.202.2.1ConstructPDVByDataSet() . . . . .	726
10.203dcm::network::NGetRQ Class Reference . . . . .	727
10.203.1Detailed Description . . . . .	727
10.203.2Member Function Documentation . . . . .	728
10.203.2.1ConstructPDV() . . . . .	728
10.204dcm::network::NGetRSP Class Reference . . . . .	728
10.204.1Detailed Description . . . . .	729
10.204.2Member Function Documentation . . . . .	729
10.204.2.1ConstructPDVByDataSet() . . . . .	729

10.205	dcm::NoEvent Class Reference	730
10.205.1	Detailed Description	730
10.206	dcm::network::NormalizedMessageFactory Class Reference	731
10.206.1	Member Function Documentation	731
10.206.1.1	ConstructNAction()	731
10.206.1.2	ConstructNCreate()	731
10.206.1.3	ConstructNDelete()	732
10.206.1.4	ConstructNEventReport()	732
10.206.1.5	ConstructNGet()	732
10.206.1.6	ConstructNSet()	732
10.207	dcm::NormalizedNetworkFunctions Class Reference	732
10.207.1	Detailed Description	733
10.207.2	Member Function Documentation	733
10.207.2.1	ConstructQuery()	733
10.207.2.2	NAction()	734
10.207.2.3	NCreate()	734
10.207.2.4	NDelete()	734
10.207.2.5	NEventReport()	734
10.207.2.6	NGet()	735
10.207.2.7	NSet()	735
10.208	dcm::network::NSetRQ Class Reference	735
10.208.1	Detailed Description	736
10.208.2	Member Function Documentation	736
10.208.2.1	ConstructPDV()	736
10.209	dcm::network::NSetRSP Class Reference	737
10.209.1	Detailed Description	737
10.209.2	Member Function Documentation	738
10.209.2.1	ConstructPDVByDataSet()	738



10.210.0dcm::Object Class Reference . . . . .	738
10.210.1Detailed Description . . . . .	740
10.210.2Constructor & Destructor Documentation . . . . .	740
10.210.2.1Object() [1/2] . . . . .	740
10.210.2.2~Object() . . . . .	740
10.210.2.3Object() [2/2] . . . . .	740
10.210.3Member Function Documentation . . . . .	741
10.210.3.1operator=() . . . . .	741
10.210.3.2Print() . . . . .	741
10.210.3.3Register() . . . . .	741
10.210.3.4UnRegister() . . . . .	741
10.210.4Friends And Related Function Documentation . . . . .	741
10.210.4.1operator<< . . . . .	742
10.210.4.2SmartPointer . . . . .	742
10.211.0dcm::OpenSSLCryptoFactory Class Reference . . . . .	742
10.211.1Constructor & Destructor Documentation . . . . .	743
10.211.1.1OpenSSLCryptoFactory() . . . . .	743
10.211.2Member Function Documentation . . . . .	743
10.211.2.1CreateCMSProvider() . . . . .	743
10.211.2.2InitOpenSSL() . . . . .	743
10.212.0dcm::OpenSSLCryptographicMessageSyntax Class Reference . . . . .	744
10.212.1Constructor & Destructor Documentation . . . . .	745
10.212.1.1OpenSSLCryptographicMessageSyntax() . . . . .	745
10.212.1.2~OpenSSLCryptographicMessageSyntax() . . . . .	745
10.212.2Member Function Documentation . . . . .	745
10.212.2.1Decrypt() . . . . .	745
10.212.2.2Encrypt() . . . . .	746
10.212.2.3GetCipherType() . . . . .	746

10.212.2.4	ParseCertificateFile()	. . . . .	746
10.212.2.5	ParseKeyFile()	. . . . .	746
10.212.2.6	SetCipherType()	. . . . .	746
10.212.2.7	SetPassword()	. . . . .	747
10.213	dcm::OpenSSLP7CryptoFactory Class Reference	. . . . .	747
10.213.1	Constructor & Destructor Documentation	. . . . .	748
10.213.1.1	OpenSSLP7CryptoFactory()	. . . . .	748
10.213.2	Member Function Documentation	. . . . .	748
10.213.2.1	CreateCMSProvider()	. . . . .	748
10.214	dcm::OpenSSLP7CryptographicMessageSyntax Class Reference	. . . . .	749
10.214.1	Detailed Description	. . . . .	750
10.214.2	Constructor & Destructor Documentation	. . . . .	750
10.214.2.1	OpenSSLP7CryptographicMessageSyntax()	. . . . .	750
10.214.2.2	~OpenSSLP7CryptographicMessageSyntax()	. . . . .	750
10.214.3	Member Function Documentation	. . . . .	750
10.214.3.1	Decrypt()	. . . . .	750
10.214.3.2	Encrypt()	. . . . .	751
10.214.3.3	GetCipherType()	. . . . .	751
10.214.3.4	ParseCertificateFile()	. . . . .	751
10.214.3.5	ParseKeyFile()	. . . . .	751
10.214.3.6	SetCipherType()	. . . . .	751
10.214.3.7	SetPassword()	. . . . .	752
10.215	dcm::Orientation Class Reference	. . . . .	752
10.215.1	Detailed Description	. . . . .	753
10.215.2	Member Enumeration Documentation	. . . . .	753
10.215.2.1	OrientationType	. . . . .	753
10.215.3	Constructor & Destructor Documentation	. . . . .	753
10.215.3.1	Orientation()	. . . . .	753

10.215.3.2~Orientation()	753
10.215.4Member Function Documentation	754
10.215.4.1GetLabel()	754
10.215.4.2GetMajorAxisFromPatientRelativeDirectionCosine()	754
10.215.4.3GetObliquityThresholdCosineValue()	754
10.215.4.4GetType()	754
10.215.4.5Print()	755
10.215.4.6SetObliquityThresholdCosineValue()	755
10.215.5Friends And Related Function Documentation	755
10.215.5.1operator<<	755
10.216dcm::Overlay Class Reference	756
10.216.1Detailed Description	758
10.216.2Member Enumeration Documentation	758
10.216.2.1OverlayType	758
10.216.3Constructor & Destructor Documentation	759
10.216.3.1Overlay() [1/2]	759
10.216.3.2~Overlay()	759
10.216.3.3Overlay() [2/2]	759
10.216.4Member Function Documentation	759
10.216.4.1Decompress()	759
10.216.4.2GetBitPosition()	760
10.216.4.3GetBitsAllocated()	760
10.216.4.4GetColumns()	760
10.216.4.5GetDescription()	760
10.216.4.6GetGroup()	760
10.216.4.7GetOrigin()	760
10.216.4.8GetOverlayData()	761
10.216.4.9GetOverlayTypeAsString()	761

10.216.4.10	GetOverlayTypeFromString()	. . . . .	761
10.216.4.10	GetRows()	. . . . .	761
10.216.4.10	GetType()	. . . . .	761
10.216.4.10	GetTypeAsEnum()	. . . . .	761
10.216.4.10	GetUnpackBuffer()	. . . . .	762
10.216.4.10	GetUnpackBufferLength()	. . . . .	762
10.216.4.10	GrabOverlayFromPixelData()	. . . . .	762
10.216.4.10	IsEmpty()	. . . . .	762
10.216.4.10	InPixelData() [1/2]	. . . . .	762
10.216.4.10	InPixelData() [2/2]	. . . . .	762
10.216.4.20	Zero()	. . . . .	763
10.216.4.20	operator=()	. . . . .	763
10.216.4.20	Print()	. . . . .	763
10.216.4.20	SetBitPosition()	. . . . .	763
10.216.4.20	SetBitsAllocated()	. . . . .	763
10.216.4.20	SetColumns()	. . . . .	764
10.216.4.20	SetDescription()	. . . . .	764
10.216.4.20	SetFrameOrigin()	. . . . .	764
10.216.4.20	SetGroup()	. . . . .	764
10.216.4.20	SetNumberOfFrames()	. . . . .	764
10.216.4.30	SetOrigin()	. . . . .	765
10.216.4.30	SetOverlay()	. . . . .	765
10.216.4.30	SetRows()	. . . . .	765
10.216.4.30	SetType()	. . . . .	765
10.216.4.30	Update()	. . . . .	765
10.217	dcm::ParseException Class Reference	. . . . .	766
10.217.1	Detailed Description	. . . . .	767
10.217.2	Constructor & Destructor Documentation	. . . . .	767

10.217.2.1	ParseException()	. . . . .	767
10.217.2.2	~ParseException()	. . . . .	767
10.217.3	Member Function Documentation	. . . . .	767
10.217.3.1	GetLastElement()	. . . . .	767
10.217.3.2	operator=()	. . . . .	767
10.217.3.3	SetLastElement()	. . . . .	768
10.218	dcm::Parser Class Reference	. . . . .	768
10.218.1	Detailed Description	. . . . .	769
10.218.2	Member Typedef Documentation	. . . . .	769
10.218.2.1	EndElementHandler	. . . . .	769
10.218.2.2	StartElementHandler	. . . . .	769
10.218.3	Member Enumeration Documentation	. . . . .	769
10.218.3.1	ErrorType	. . . . .	769
10.218.4	Constructor & Destructor Documentation	. . . . .	770
10.218.4.1	Parser()	. . . . .	770
10.218.4.2	~Parser()	. . . . .	770
10.218.5	Member Function Documentation	. . . . .	770
10.218.5.1	GetBuffer()	. . . . .	770
10.218.5.2	GetCurrentByteIndex()	. . . . .	770
10.218.5.3	GetErrorCode()	. . . . .	771
10.218.5.4	GetErrorString()	. . . . .	771
10.218.5.5	GetUserData()	. . . . .	771
10.218.5.6	Parse()	. . . . .	771
10.218.5.7	ParseBuffer()	. . . . .	771
10.218.5.8	Process()	. . . . .	771
10.218.5.9	SetElementHandler()	. . . . .	772
10.218.5.10	SetUserData()	. . . . .	772
10.219	dcm::Patient Class Reference	. . . . .	772

10.219.1	Detailed Description	. 772
10.219.2	Constructor & Destructor Documentation	. 772
10.219.2.1	IPatient()	. 772
10.220	dcm::network::PDataTFPDU Class Reference	. 773
10.220.1	Detailed Description	. 774
10.220.2	Member Typedef Documentation	. 774
10.220.2.1	SizeType	. 774
10.220.3	Constructor & Destructor Documentation	. 774
10.220.3.1	IPDataTFPDU()	. 774
10.220.4	Member Function Documentation	. 774
10.220.4.1	AddPresentationDataValue()	. 774
10.220.4.2	GetNumberOfPresentationDataValues()	. 775
10.220.4.3	GetPresentationDataValue()	. 775
10.220.4.4	IsLastFragment()	. 775
10.220.4.5	Print()	. 775
10.220.4.6	Read()	. 775
10.220.4.7	ReadInto()	. 775
10.220.4.8	Size()	. 776
10.220.4.9	Write()	. 776
10.221	dcm::PDBElement Class Reference	. 776
10.221.1	Detailed Description	. 777
10.221.2	Constructor & Destructor Documentation	. 777
10.221.2.1	IPDBElement()	. 777
10.221.3	Member Function Documentation	. 777
10.221.3.1	GetName()	. 778
10.221.3.2	GetValue()	. 778
10.221.3.3	operator==( )	. 778
10.221.3.4	SetName()	. 778

10.221.3.5	SetValue()	. . . . .	778
10.221.4	Friends And Related Function Documentation	. . . . .	778
10.221.4.1	operator<<	. . . . .	779
10.221.5	Member Data Documentation	. . . . .	779
10.221.5.1	NameField	. . . . .	779
10.221.5.2	ValueField	. . . . .	779
10.222	dcm::PDBHeader Class Reference	. . . . .	779
10.222.1	Detailed Description	. . . . .	780
10.222.2	Constructor & Destructor Documentation	. . . . .	780
10.222.2.1	PDBHeader()	. . . . .	780
10.222.2.2	~PDBHeader()	. . . . .	780
10.222.3	Member Function Documentation	. . . . .	781
10.222.3.1	FindPDBElementByName()	. . . . .	781
10.222.3.2	GetPDBEEnd()	. . . . .	781
10.222.3.3	GetPDBElementByName()	. . . . .	781
10.222.3.4	GetPDBInfoTag()	. . . . .	781
10.222.3.5	LoadFromDataElement()	. . . . .	782
10.222.3.6	Print()	. . . . .	782
10.222.4	Friends And Related Function Documentation	. . . . .	782
10.222.4.1	operator<<	. . . . .	782
10.223	dcm::PDFCodec Class Reference	. . . . .	783
10.223.1	Detailed Description	. . . . .	784
10.223.2	Constructor & Destructor Documentation	. . . . .	784
10.223.2.1	PDFCodec()	. . . . .	784
10.223.2.2	~PDFCodec()	. . . . .	784
10.223.3	Member Function Documentation	. . . . .	784
10.223.3.1	CanCode()	. . . . .	784
10.223.3.2	CanDecode()	. . . . .	785

10.223.3.3	Decode()	785
10.224	dcm::network::PDUFactory Class Reference	785
10.224.1	Detailed Description	786
10.224.2	Member Function Documentation	786
10.224.2.1	ConstructAbortPDU()	786
10.224.2.2	ConstructPDU()	786
10.224.2.3	ConstructReleasePDU()	786
10.224.2.4	CreateCEchoPDU()	787
10.224.2.5	CreateCFindPDU()	787
10.224.2.6	CreateCMovePDU()	787
10.224.2.7	CreateCStoreRQPDU()	787
10.224.2.8	CreateCStoreRSPPDU()	787
10.224.2.9	CreateNActionPDU()	787
10.224.2.10	CreateNCreatePDU()	788
10.224.2.11	CreateNDeletePDU()	788
10.224.2.12	CreateNEventReportPDU()	788
10.224.2.13	CreateNGetPDU()	788
10.224.2.14	CreateNSetPDU()	788
10.224.2.15	DetermineEventByPDU()	788
10.224.2.16	GetPDVs()	789
10.225	dcm::PersonName Class Reference	789
10.225.1	Detailed Description	789
10.225.2	Member Function Documentation	789
10.225.2.1	GetMaxLength()	790
10.225.2.2	GetNumberOfComponents()	790
10.225.2.3	Print()	790
10.225.2.4	SetBlob()	790
10.225.2.5	SetComponents() [1/2]	790



10.225.2.6SetComponents() [2/2]	790
10.225.3Member Data Documentation	791
10.225.3.1Component	791
10.225.3.2MaxLength	791
10.225.3.3MaxNumberOfComponents	791
10.225.3.4Padding	791
10.225.3.5Separator	791
10.226dcm::PGXCodec Class Reference	792
10.226.1Detailed Description	793
10.226.2Constructor & Destructor Documentation	793
10.226.2.1PGXCodec()	793
10.226.2.2~PGXCodec()	793
10.226.3Member Function Documentation	793
10.226.3.1CanCode()	793
10.226.3.2CanDecode()	794
10.226.3.3Clone()	794
10.226.3.4GetHeaderInfo()	794
10.226.3.5Read()	794
10.226.3.6Write()	794
10.227dcm::PhotometricInterpretation Class Reference	795
10.227.1Detailed Description	796
10.227.2Member Enumeration Documentation	796
10.227.2.1PIType	796
10.227.3Constructor & Destructor Documentation	796
10.227.3.1PhotometricInterpretation()	796
10.227.4Member Function Documentation	797
10.227.4.1GetPIString()	797
10.227.4.2GetPIType()	797

10.227.4.3	GetSamplesPerPixel()	797
10.227.4.4	GetString()	797
10.227.4.5	GetType()	797
10.227.4.6	IsLossless()	797
10.227.4.7	IsLossy()	798
10.227.4.8	IsRetired()	798
10.227.4.9	IsSameColorSpace()	798
10.227.4.10	operator PType()	798
10.227.5	Friends And Related Function Documentation	798
10.227.5.1	operator <<	798
10.228	dcm::PixelFormat Class Reference	798
10.228.1	Detailed Description	800
10.228.2	Member Enumeration Documentation	800
10.228.2.1	ScalarType	800
10.228.3	Constructor & Destructor Documentation	801
10.228.3.1	PixelFormat() [1/2]	801
10.228.3.2	PixelFormat() [2/2]	801
10.228.4	Member Function Documentation	801
10.228.4.1	GetBitsAllocated()	802
10.228.4.2	GetBitsStored()	802
10.228.4.3	GetHighBit()	802
10.228.4.4	GetMax()	802
10.228.4.5	GetMin()	802
10.228.4.6	GetPixelRepresentation()	803
10.228.4.7	GetPixelSize()	803
10.228.4.8	GetSamplesPerPixel()	803
10.228.4.9	GetScalarType()	803
10.228.4.10	GetScalarTypeAsString()	804

10.228.4.11IsCompatible()	804
10.228.4.12Valid()	804
10.228.4.13operator ScalarType()	804
10.228.4.14operator"!="() [1/2]	804
10.228.4.15operator"!="() [2/2]	804
10.228.4.16operator==( ) [1/2]	805
10.228.4.17operator==( ) [2/2]	805
10.228.4.18Print()	805
10.228.4.19SetBitsAllocated()	805
10.228.4.20SetBitsStored()	805
10.228.4.21SetHighBit()	805
10.228.4.22SetPixelRepresentation()	806
10.228.4.23SetSamplesPerPixel()	806
10.228.4.24SetScalarType()	806
10.228.4.25Validate()	806
10.228.5Friends And Related Function Documentation	807
10.228.5.1Bitmap	807
10.228.5.2operator<<	807
10.229cdcm::Pixmap Class Reference	807
10.229.1Detailed Description	809
10.229.2Constructor & Destructor Documentation	809
10.229.2.1Pixmap()	809
10.229.2.2~Pixmap()	809
10.229.3Member Function Documentation	809
10.229.3.1AreOverlaysInPixelData()	809
10.229.3.2GetCurve() [1/2]	810
10.229.3.3GetCurve() [2/2]	810
10.229.3.4GetIconImage() [1/2]	810

10.229.3.5	GetIconImage() [2/2]	810
10.229.3.6	GetNumberOfCurves()	810
10.229.3.7	GetNumberOfOverlays()	810
10.229.3.8	GetOverlay() [1/2]	811
10.229.3.9	GetOverlay() [2/2]	811
10.229.3.10	Print()	811
10.229.3.11	RemoveOverlay()	811
10.229.3.12	SetIconImage()	811
10.229.3.13	SetNumberOfCurves()	811
10.229.3.14	SetNumberOfOverlays()	812
10.229.3.15	UnusedBitsPresentInPixelData()	812
10.229.4	Member Data Documentation	812
10.229.4.1	Curves	812
10.229.4.2	Icon	812
10.229.4.3	Overlays	812
10.230	dcm::PixmapReader Class Reference	813
10.230.1	Detailed Description	815
10.230.2	Constructor & Destructor Documentation	815
10.230.2.1	PixmapReader()	815
10.230.2.2	~PixmapReader()	815
10.230.3	Member Function Documentation	815
10.230.3.1	GetPixmap() [1/2]	815
10.230.3.2	GetPixmap() [2/2]	816
10.230.3.3	Read()	816
10.230.3.4	ReadACRNEMAImage()	816
10.230.3.5	ReadImage()	816
10.230.3.6	ReadImageInternal()	816
10.230.4	Member Data Documentation	817

10.230.4.1PixelData . . . . .	817
10.231dcm::PixmapToPixmapFilter Class Reference . . . . .	817
10.231.1Detailed Description . . . . .	818
10.231.2Constructor & Destructor Documentation . . . . .	819
10.231.2.1PixmapToPixmapFilter() . . . . .	819
10.231.2.2~PixmapToPixmapFilter() . . . . .	819
10.231.3Member Function Documentation . . . . .	819
10.231.3.1GetInput() . . . . .	819
10.231.3.2GetOutput() . . . . .	819
10.231.3.3GetOutputAsPixmap() . . . . .	819
10.232dcm::PixmapWriter Class Reference . . . . .	820
10.232.1Detailed Description . . . . .	822
10.232.2Constructor & Destructor Documentation . . . . .	822
10.232.2.1PixmapWriter() . . . . .	822
10.232.2.2~PixmapWriter() . . . . .	822
10.232.3Member Function Documentation . . . . .	822
10.232.3.1DoIconImage() . . . . .	823
10.232.3.2GetImage() [1/2] . . . . .	823
10.232.3.3GetImage() [2/2] . . . . .	823
10.232.3.4GetPixmap() [1/2] . . . . .	823
10.232.3.5GetPixmap() [2/2] . . . . .	823
10.232.3.6PrepareWrite() . . . . .	823
10.232.3.7SetImage() . . . . .	824
10.232.3.8SetPixmap() . . . . .	824
10.232.3.9Write() . . . . .	824
10.232.4Member Data Documentation . . . . .	824
10.232.4.1PixelData . . . . .	824
10.233dcm::PNMCodec Class Reference . . . . .	825

10.233.1	Detailed Description	826
10.233.2	Constructor & Destructor Documentation	826
10.233.2.1	PNMCodec()	826
10.233.2.2	~PNMCodec()	826
10.233.3	Member Function Documentation	826
10.233.3.1	CanCode()	827
10.233.3.2	CanDecode()	827
10.233.3.3	Clone()	827
10.233.3.4	GetBufferLength()	827
10.233.3.5	GetHeaderInfo()	827
10.233.3.6	Read()	828
10.233.3.7	SetBufferLength()	828
10.233.3.8	Write()	828
10.234	dcm::Preamble Class Reference	828
10.234.1	Detailed Description	829
10.234.2	Constructor & Destructor Documentation	829
10.234.2.1	Preamble() [1/2]	830
10.234.2.2	~Preamble()	830
10.234.2.3	Preamble() [2/2]	830
10.234.3	Member Function Documentation	830
10.234.3.1	Clear()	830
10.234.3.2	Create()	830
10.234.3.3	GetInternal()	830
10.234.3.4	GetLength()	831
10.234.3.5	IsEmpty()	831
10.234.3.6	IsValid()	831
10.234.3.7	operator=()	831
10.234.3.8	Print()	831

10.234.3.9Read()	831
10.234.3.10Remove()	832
10.234.3.11Valid()	832
10.234.3.12Write()	832
10.234.4Friends And Related Function Documentation	832
10.234.4.1operator<<	832
10.235dcm::PresentationContext Class Reference	833
10.235.1Detailed Description	834
10.235.2Member Typedef Documentation	834
10.235.2.1SizeType	834
10.235.2.2TransferSyntaxArrayType	834
10.235.3Constructor & Destructor Documentation	834
10.235.3.1PresentationContext() [1/2]	834
10.235.3.2PresentationContext() [2/2]	835
10.235.4Member Function Documentation	835
10.235.4.1AddTransferSyntax()	835
10.235.4.2GetAbstractSyntax()	835
10.235.4.3GetNumberOfTransferSyntaxes()	835
10.235.4.4GetPresentationContextID()	835
10.235.4.5GetTransferSyntax()	835
10.235.4.6operator==( )	836
10.235.4.7Print()	836
10.235.4.8SetAbstractSyntax()	836
10.235.4.9SetPresentationContextID()	836
10.235.5Member Data Documentation	836
10.235.5.1AbstractSyntax	836
10.235.5.2ID	836
10.235.5.3TransferSyntaxes	837

10.236.0	dcm::network::PresentationContextAC Class Reference . . . . .	837
10.236.1	Detailed Description . . . . .	837
10.236.2	Constructor & Destructor Documentation . . . . .	837
10.236.2.1	PresentationContextAC() . . . . .	838
10.236.3	Member Function Documentation . . . . .	838
10.236.3.1	GetPresentationContextID() . . . . .	838
10.236.3.2	GetReason() . . . . .	838
10.236.3.3	GetTransferSyntax() . . . . .	838
10.236.3.4	Print() . . . . .	838
10.236.3.5	Read() . . . . .	838
10.236.3.6	SetPresentationContextID() . . . . .	839
10.236.3.7	SetReason() . . . . .	839
10.236.3.8	SetTransferSyntax() . . . . .	839
10.236.3.9	Size() . . . . .	839
10.236.3.10	Write() . . . . .	839
10.237.0	dcm::PresentationContextGenerator Class Reference . . . . .	839
10.237.1	Detailed Description . . . . .	840
10.237.2	Member Typedef Documentation . . . . .	841
10.237.2.1	PresentationContextArrayType . . . . .	841
10.237.2.2	SizeType . . . . .	841
10.237.3	Constructor & Destructor Documentation . . . . .	841
10.237.3.1	PresentationContextGenerator() . . . . .	841
10.237.4	Member Function Documentation . . . . .	841
10.237.4.1	AddFromFile() . . . . .	841
10.237.4.2	AddPresentationContext() . . . . .	841
10.237.4.3	GenerateFromFilenames() . . . . .	842
10.237.4.4	GenerateFromUID() . . . . .	842
10.237.4.5	GetDefaultTransferSyntax() . . . . .	842



10.237.4.6GetPresentationContexts()	842
10.237.4.7SetDefaultTransferSyntax()	842
10.237.4.8SetMergeModeToAbstractSyntax()	843
10.237.4.9SetMergeModeToTransferSyntax()	843
10.238.0dcm::network::PresentationContextRQ Class Reference	843
10.238.1Detailed Description	844
10.238.2Member Typedef Documentation	844
10.238.2.1SizeType	844
10.238.3Constructor & Destructor Documentation	844
10.238.3.1PresentationContextRQ() [1/3]	844
10.238.3.2PresentationContextRQ() [2/3]	844
10.238.3.3PresentationContextRQ() [3/3]	844
10.238.4Member Function Documentation	845
10.238.4.1AddTransferSyntax()	845
10.238.4.2GetAbstractSyntax() [1/2]	845
10.238.4.3GetAbstractSyntax() [2/2]	845
10.238.4.4GetNumberOfTransferSyntaxes()	845
10.238.4.5GetPresentationContextID()	845
10.238.4.6GetTransferSyntax() [1/2]	845
10.238.4.7GetTransferSyntax() [2/2]	846
10.238.4.8GetTransferSyntaxes()	846
10.238.4.9operator==()	846
10.238.4.10Print()	846
10.238.4.11Read()	846
10.238.4.12SetAbstractSyntax()	846
10.238.4.13SetPresentationContextID()	847
10.238.4.14Size()	847
10.238.4.15Write()	847

10.239	dcm::network::PresentationDataValue Class Reference . . . . .	847
10.239.1	Detailed Description . . . . .	848
10.239.2	Constructor & Destructor Documentation . . . . .	848
10.239.2.1	PresentationDataValue() . . . . .	848
10.239.3	Member Function Documentation . . . . .	848
10.239.3.1	ConcatenatePDVBlobs() . . . . .	848
10.239.3.2	ConcatenatePDVBlobsAsExplicit() . . . . .	848
10.239.3.3	GetBlob() . . . . .	849
10.239.3.4	GetIsCommand() . . . . .	849
10.239.3.5	GetIsLastFragment() . . . . .	849
10.239.3.6	GetMessageHeader() . . . . .	849
10.239.3.7	GetPresentationContextID() . . . . .	849
10.239.3.8	Print() . . . . .	849
10.239.3.9	Read() . . . . .	849
10.239.3.10	ReadInto() . . . . .	850
10.239.3.11	SetBlob() . . . . .	850
10.239.3.12	SetCommand() . . . . .	850
10.239.3.13	SetDataSet() . . . . .	850
10.239.3.14	SetLastFragment() . . . . .	850
10.239.3.15	SetMessageHeader() . . . . .	851
10.239.3.16	SetPresentationContextID() . . . . .	851
10.239.3.17	Size() . . . . .	851
10.239.3.18	Write() . . . . .	851
10.240	dcm::Printer Class Reference . . . . .	851
10.240.1	Detailed Description . . . . .	853
10.240.2	Member Enumeration Documentation . . . . .	853
10.240.2.1	PrintStyles . . . . .	853
10.240.3	Constructor & Destructor Documentation . . . . .	853

10.240.3.1Printer()	853
10.240.3.2~Printer()	854
10.240.4Member Function Documentation	854
10.240.4.1GetPrintStyle()	854
10.240.4.2Print()	854
10.240.4.3PrintDataElement()	854
10.240.4.4PrintDataSet()	855
10.240.4.5PrintSQ()	855
10.240.4.6SetColor()	855
10.240.4.7SetFile()	855
10.240.4.8SetStyle()	856
10.240.5Member Data Documentation	856
10.240.5.1F	856
10.240.5.2MaxPrintLength	856
10.240.5.3PrintStyle	856
10.241dcm::PrivateDict Class Reference	856
10.241.1Detailed Description	857
10.241.2Constructor & Destructor Documentation	857
10.241.2.1PrivateDict()	857
10.241.2.2~PrivateDict()	857
10.241.3Member Function Documentation	857
10.241.3.1AddDictEntry()	858
10.241.3.2FindDictEntry()	858
10.241.3.3GetDictEntry()	858
10.241.3.4IsEmpty()	858
10.241.3.5LoadDefault()	858
10.241.3.6PrintXML()	858
10.241.3.7RemoveDictEntry()	859

10.241.4	Friends And Related Function Documentation . . . . .	859
10.241.4.1	Dicts . . . . .	859
10.241.4.2	operator<< . . . . .	859
10.242.0	dcm::PrivateTag Class Reference . . . . .	859
10.242.1	Detailed Description . . . . .	860
10.242.2	Constructor & Destructor Documentation . . . . .	860
10.242.2.1	PrivateTag() [1/2] . . . . .	861
10.242.2.2	PrivateTag() [2/2] . . . . .	861
10.242.3	Member Function Documentation . . . . .	861
10.242.3.1	GetAsDataElement() . . . . .	861
10.242.3.2	GetOwner() . . . . .	861
10.242.3.3	operator<() . . . . .	861
10.242.3.4	ReadFromCommaSeparatedString() . . . . .	862
10.242.3.5	SetOwner() . . . . .	862
10.242.4	Friends And Related Function Documentation . . . . .	862
10.242.4.1	operator<< . . . . .	862
10.243.0	dcm::ProgressEvent Class Reference . . . . .	862
10.243.1	Detailed Description . . . . .	863
10.243.2	Member Typedef Documentation . . . . .	864
10.243.2.1	Self . . . . .	864
10.243.2.2	Superclass . . . . .	864
10.243.3	Constructor & Destructor Documentation . . . . .	864
10.243.3.1	ProgressEvent() [1/2] . . . . .	864
10.243.3.2	~ProgressEvent() . . . . .	864
10.243.3.3	ProgressEvent() [2/2] . . . . .	864
10.243.4	Member Function Documentation . . . . .	864
10.243.4.1	CheckEvent() . . . . .	865
10.243.4.2	GetEventName() . . . . .	865

10.243.4.3	GetProgress()	865
10.243.4.4	MakeObject()	865
10.243.4.5	SetProgress()	865
10.244	dcm::PVRGCodec Class Reference	866
10.244.1	Detailed Description	867
10.244.2	Constructor & Destructor Documentation	867
10.244.2.1	PVRGCodec()	867
10.244.2.2	~PVRGCodec()	867
10.244.3	Member Function Documentation	867
10.244.3.1	CanCode()	868
10.244.3.2	CanDecode()	868
10.244.3.3	Clone()	868
10.244.3.4	Code()	868
10.244.3.5	Decode()	869
10.244.3.6	SetLossyFlag()	869
10.245	dcm::PythonFilter Class Reference	869
10.245.1	Detailed Description	869
10.245.2	Constructor & Destructor Documentation	870
10.245.2.1	PythonFilter()	870
10.245.2.2	~PythonFilter()	870
10.245.3	Member Function Documentation	870
10.245.3.1	GetFile() [1/2]	870
10.245.3.2	GetFile() [2/2]	870
10.245.3.3	SetDicts()	870
10.245.3.4	SetFile()	870
10.245.3.5	ToPyObject()	871
10.245.3.6	UseDictAlways()	871
10.246	dcm::QueryBase Class Reference	871

10.246.1	Detailed Description	872
10.246.2	Constructor & Destructor Documentation	872
10.246.2.1	~QueryBase()	872
10.246.3	Member Function Documentation	872
10.246.3.1	GetAllRequiredTags()	872
10.246.3.2	GetAllTags()	873
10.246.3.3	GetHierachicalSearchTags()	873
10.246.3.4	GetName()	873
10.246.3.5	GetOptionalTags()	873
10.246.3.6	GetQueryLevel()	873
10.246.3.7	GetRequiredTags()	874
10.246.3.8	GetUniqueTags()	874
10.247	dcm::QueryFactory Class Reference	874
10.247.1	Detailed Description	874
10.247.2	Member Function Documentation	875
10.247.2.1	GetCharacterFromCurrentLocale()	875
10.247.2.2	ListCharSets()	875
10.247.2.3	ProduceCharacterSetDataElement()	875
10.247.2.4	ProduceQuery() [1/2]	875
10.247.2.5	ProduceQuery() [2/2]	876
10.248	dcm::QueryImage Class Reference	876
10.248.1	Detailed Description	877
10.248.2	Member Function Documentation	877
10.248.2.1	GetHierachicalSearchTags()	877
10.248.2.2	GetName()	877
10.248.2.3	GetOptionalTags()	877
10.248.2.4	GetQueryLevel()	878
10.248.2.5	GetRequiredTags()	878

10.248.2.6GetUniqueTags()	878
10.249dcm::QueryPatient Class Reference	878
10.249.1Detailed Description	879
10.249.2Member Function Documentation	879
10.249.2.1GetHierachicalSearchTags()	879
10.249.2.2GetName()	880
10.249.2.3GetOptionalTags()	880
10.249.2.4GetQueryLevel()	880
10.249.2.5GetRequiredTags()	880
10.249.2.6GetUniqueTags()	880
10.250dcm::QuerySeries Class Reference	881
10.250.1Detailed Description	882
10.250.2Member Function Documentation	882
10.250.2.1GetHierachicalSearchTags()	882
10.250.2.2GetName()	882
10.250.2.3GetOptionalTags()	882
10.250.2.4GetQueryLevel()	882
10.250.2.5GetRequiredTags()	883
10.250.2.6GetUniqueTags()	883
10.251dcm::QueryStudy Class Reference	883
10.251.1Detailed Description	884
10.251.2Member Function Documentation	884
10.251.2.1GetHierachicalSearchTags()	884
10.251.2.2GetName()	885
10.251.2.3GetOptionalTags()	885
10.251.2.4GetQueryLevel()	885
10.251.2.5GetRequiredTags()	885
10.251.2.6GetUniqueTags()	885

10.252.0 dcm::RAWCodec Class Reference . . . . .	886
10.252.1 Detailed Description . . . . .	887
10.252.2 Constructor & Destructor Documentation . . . . .	887
10.252.2.1 RAWCodec() . . . . .	887
10.252.2.2 ~RAWCodec() . . . . .	887
10.252.3 Member Function Documentation . . . . .	887
10.252.3.1 CanCode() . . . . .	888
10.252.3.2 CanDecode() . . . . .	888
10.252.3.3 Clone() . . . . .	888
10.252.3.4 Code() . . . . .	888
10.252.3.5 Decode() . . . . .	889
10.252.3.6 DecodeByStreams() . . . . .	889
10.252.3.7 DecodeBytes() . . . . .	889
10.252.3.8 GetHeaderInfo() . . . . .	889
10.253.0 dcm::Reader Class Reference . . . . .	890
10.253.1 Detailed Description . . . . .	892
10.253.2 Constructor & Destructor Documentation . . . . .	893
10.253.2.1 Reader() . . . . .	893
10.253.2.2 ~Reader() . . . . .	893
10.253.3 Member Function Documentation . . . . .	893
10.253.3.1 CanRead() . . . . .	893
10.253.3.2 GetFile() [1/2] . . . . .	893
10.253.3.3 GetFile() [2/2] . . . . .	894
10.253.3.4 GetStreamCurrentPosition() . . . . .	894
10.253.3.5 GetStreamPtr() . . . . .	894
10.253.3.6 Read() . . . . .	894
10.253.3.7 ReadDataSet() . . . . .	895
10.253.3.8 ReadMetaInformation() . . . . .	895



10.253.3.9	<a href="#">ReadPreamble()</a> . . . . .	895
10.253.3.10	<a href="#">ReadSelectedPrivateTags()</a> . . . . .	895
10.253.3.11	<a href="#">ReadSelectedTags()</a> . . . . .	895
10.253.3.12	<a href="#">ReadUpToTag()</a> . . . . .	895
10.253.3.13	<a href="#">SetFile()</a> . . . . .	896
10.253.3.14	<a href="#">SetFileName()</a> . . . . .	896
10.253.3.15	<a href="#">SetStream()</a> . . . . .	896
10.253.4	<a href="#">Friends And Related Function Documentation</a> . . . . .	897
10.253.4.1	<a href="#">StreamImageReader</a> . . . . .	897
10.253.5	<a href="#">Member Data Documentation</a> . . . . .	897
10.253.5.1	<a href="#">F</a> . . . . .	897
10.254	<a href="#">dcm::RealWorldValueMappingContent Struct Reference</a> . . . . .	897
10.254.1	<a href="#">Member Data Documentation</a> . . . . .	898
10.254.1.1	<a href="#">CodeMeaning</a> . . . . .	898
10.254.1.2	<a href="#">CodeValue</a> . . . . .	898
10.254.1.3	<a href="#">RealWorldValueIntercept</a> . . . . .	898
10.254.1.4	<a href="#">RealWorldValueSlope</a> . . . . .	898
10.255	<a href="#">dcm::Region Class Reference</a> . . . . .	899
10.255.1	<a href="#">Detailed Description</a> . . . . .	899
10.255.2	<a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	899
10.255.2.1	<a href="#">Region()</a> . . . . .	900
10.255.2.2	<a href="#">~Region()</a> . . . . .	900
10.255.3	<a href="#">Member Function Documentation</a> . . . . .	900
10.255.3.1	<a href="#">Area()</a> . . . . .	900
10.255.3.2	<a href="#">Clone()</a> . . . . .	900
10.255.3.3	<a href="#">ComputeBoundingBox()</a> . . . . .	900
10.255.3.4	<a href="#">Empty()</a> . . . . .	901
10.255.3.5	<a href="#">IsValid()</a> . . . . .	901

10.255.3.6Print()	901
10.256dcm::Rescaler Class Reference	901
10.256.1Detailed Description	902
10.256.2Constructor & Destructor Documentation	903
10.256.2.1Rescaler()	903
10.256.2.2~Rescaler()	903
10.256.3Member Function Documentation	903
10.256.3.1ComputeInterceptSlopePixelType()	903
10.256.3.2ComputePixelTypeFromMinMax()	904
10.256.3.3GetIntercept()	904
10.256.3.4GetSlope()	904
10.256.3.5InverseRescale()	904
10.256.3.6InverseRescaleFunctionIntoBestFit()	904
10.256.3.7Rescale()	904
10.256.3.8RescaleFunctionIntoBestFit()	905
10.256.3.9SetIntercept()	905
10.256.3.10SetMinMaxForPixelType()	905
10.256.3.11SetPixelFormat()	905
10.256.3.12SetSlope()	905
10.256.3.13SetTargetPixelType()	906
10.256.3.14SetUseTargetPixelType()	906
10.257dcm::RLECodec Class Reference	906
10.257.1Detailed Description	908
10.257.2Constructor & Destructor Documentation	908
10.257.2.1RLECodec()	908
10.257.2.2~RLECodec()	908
10.257.3Member Function Documentation	908
10.257.3.1AppendFrameEncode()	908

10.257.3.2AppendRowEncode()	909
10.257.3.3CanCode()	909
10.257.3.4CanDecode()	909
10.257.3.5Clone()	909
10.257.3.6Code()	909
10.257.3.7Decode()	910
10.257.3.8DecodeByStreams()	910
10.257.3.9DecodeExtent()	910
10.257.3.10GetBufferLength()	910
10.257.3.11GetHeaderInfo()	910
10.257.3.12FrameEncoder()	911
10.257.3.13RowEncoder()	911
10.257.3.14SetBufferLength()	911
10.257.3.15SetLength()	911
10.257.3.16StartEncode()	911
10.257.3.17StopEncode()	911
10.257.4Friends And Related Function Documentation	912
10.257.4.1ImageRegionReader	912
10.258dcm::network::RoleSelectionSub Class Reference	912
10.258.1Detailed Description	912
10.258.2Constructor & Destructor Documentation	912
10.258.2.1RoleSelectionSub()	912
10.258.3Member Function Documentation	913
10.258.3.1Print()	913
10.258.3.2Read()	913
10.258.3.3SetTuple()	913
10.258.3.4Size()	913
10.258.3.5Write()	913

10.259	dcm::SerieHelper::Rule Struct Reference . . . . .	914
10.259.1	Member Data Documentation . . . . .	914
10.259.1.1	elem . . . . .	914
10.259.1.2	group . . . . .	915
10.259.1.3	op . . . . .	915
10.259.1.4	value . . . . .	915
10.260	dcm::Scanner Class Reference . . . . .	915
10.260.1	Detailed Description . . . . .	917
10.260.2	Member Typedef Documentation . . . . .	918
10.260.2.1	ConstIterator . . . . .	918
10.260.2.2	MappingType . . . . .	918
10.260.2.3	TagToValue . . . . .	918
10.260.2.4	TagToValueValueType . . . . .	919
10.260.2.5	ValuesType . . . . .	919
10.260.3	Constructor & Destructor Documentation . . . . .	919
10.260.3.1	Scanner() . . . . .	919
10.260.3.2	~Scanner() . . . . .	919
10.260.4	Member Function Documentation . . . . .	919
10.260.4.1	AddPrivateTag() . . . . .	919
10.260.4.2	AddSkipTag() . . . . .	919
10.260.4.3	AddTag() . . . . .	920
10.260.4.4	Begin() . . . . .	920
10.260.4.5	ClearSkipTags() . . . . .	920
10.260.4.6	ClearTags() . . . . .	920
10.260.4.7	End() . . . . .	920
10.260.4.8	GetAllFileNamesFromTagToValue() . . . . .	920
10.260.4.9	GetFilenameFromTagToValue() . . . . .	921
10.260.4.10	GetFileNames() . . . . .	921

10.260.4.10	GetKeys()	921
10.260.4.10	GetMapping()	921
10.260.4.10	GetMappingFromTagToValue()	921
10.260.4.10	GetMappings()	922
10.260.4.10	GetOrderedValues()	922
10.260.4.10	GetValue()	922
10.260.4.10	GetValues() [1/2]	922
10.260.4.10	GetValues() [2/2]	923
10.260.4.10	Key()	923
10.260.4.20	New()	923
10.260.4.20	Print()	923
10.260.4.20	ProcessPublicTag()	924
10.260.4.20	Scan()	924
10.260.5	Friends And Related Function Documentation	924
10.260.5.10	operator<<	924
10.260.6	dcm::Segment Class Reference	925
10.261.1	Detailed Description	927
10.261.2	Member Typedef Documentation	927
10.261.2.10	BasicCodedEntryVector	927
10.261.2.20	SurfaceVector	927
10.261.3	Member Enumeration Documentation	927
10.261.3.10	ALGOType	927
10.261.4	Constructor & Destructor Documentation	928
10.261.4.10	Segment()	928
10.261.4.20	~Segment()	928
10.261.5	Member Function Documentation	928
10.261.5.10	AddSurface()	928
10.261.5.20	GetALGOType()	928

10.261.5.3GetALGOTypeString()	929
10.261.5.4GetAnatomicRegion() [1/2]	929
10.261.5.5GetAnatomicRegion() [2/2]	929
10.261.5.6GetAnatomicRegionModifiers() [1/2]	929
10.261.5.7GetAnatomicRegionModifiers() [2/2]	929
10.261.5.8GetPropertyCategory() [1/2]	929
10.261.5.9GetPropertyCategory() [2/2]	929
10.261.5.10GetPropertyType() [1/2]	930
10.261.5.10GetPropertyType() [2/2]	930
10.261.5.10GetPropertyTypeModifiers() [1/2]	930
10.261.5.10GetPropertyTypeModifiers() [2/2]	930
10.261.5.10GetSegmentAlgorithmName()	930
10.261.5.10GetSegmentAlgorithmType()	930
10.261.5.10GetSegmentDescription()	930
10.261.5.10GetSegmentLabel()	931
10.261.5.10GetSegmentNumber()	931
10.261.5.10GetSurface()	931
10.261.5.20GetSurfaceCount()	931
10.261.5.20GetSurfaces() [1/2]	931
10.261.5.20GetSurfaces() [2/2]	931
10.261.5.20SetAnatomicRegion()	931
10.261.5.20SetAnatomicRegionModifiers()	932
10.261.5.20SetPropertyCategory()	932
10.261.5.20SetPropertyType()	932
10.261.5.20SetPropertyTypeModifiers()	932
10.261.5.20SetSegmentAlgorithmName()	932
10.261.5.20SetSegmentAlgorithmType() [1/2]	932
10.261.5.30SetSegmentAlgorithmType() [2/2]	933

10.261.5.3SetSegmentDescription()	933
10.261.5.3SetSegmentLabel()	933
10.261.5.3SetSegmentNumber()	933
10.261.5.3SetSurfaceCount()	933
10.261.6Member Data Documentation	933
10.261.6.1AnatomicRegion	933
10.261.6.2AnatomicRegionModifiers	934
10.261.6.3PropertyCategory	934
10.261.6.4PropertyType	934
10.261.6.5PropertyTypeModifiers	934
10.261.6.6SegmentAlgorithmName	934
10.261.6.7SegmentAlgorithmType	934
10.261.6.8SegmentDescription	934
10.261.6.9SegmentLabel	935
10.261.6.10SegmentNumber	935
10.261.6.11SurfaceCount	935
10.261.6.12Surfaces	935
10.262dcm::SegmentedPaletteColorLookupTable Class Reference	935
10.262.1Detailed Description	936
10.262.2Constructor & Destructor Documentation	936
10.262.2.1SegmentedPaletteColorLookupTable()	936
10.262.2.2~SegmentedPaletteColorLookupTable()	937
10.262.3Member Function Documentation	937
10.262.3.1Print()	937
10.262.3.2SetLUT()	937
10.263dcm::SegmentReader Class Reference	938
10.263.1Detailed Description	939
10.263.2Member Typedef Documentation	939

10.263.2.1SegmentMap . . . . .	940
10.263.2.2SegmentVector . . . . .	940
10.263.3Constructor & Destructor Documentation . . . . .	940
10.263.3.1SegmentReader() . . . . .	940
10.263.3.2~SegmentReader() . . . . .	940
10.263.4Member Function Documentation . . . . .	940
10.263.4.1GetSegments() [1/2] . . . . .	940
10.263.4.2GetSegments() [2/2] . . . . .	940
10.263.4.3Read() . . . . .	941
10.263.4.4ReadSegment() . . . . .	941
10.263.4.5ReadSegments() . . . . .	941
10.263.5Member Data Documentation . . . . .	941
10.263.5.1Segments . . . . .	941
10.264dcm::SegmentWriter Class Reference . . . . .	942
10.264.1Detailed Description . . . . .	943
10.264.2Member Typedef Documentation . . . . .	943
10.264.2.1SegmentVector . . . . .	943
10.264.3Constructor & Destructor Documentation . . . . .	943
10.264.3.1SegmentWriter() . . . . .	943
10.264.3.2~SegmentWriter() . . . . .	943
10.264.4Member Function Documentation . . . . .	944
10.264.4.1AddSegment() . . . . .	944
10.264.4.2GetNumberOfSegments() . . . . .	944
10.264.4.3GetSegment() . . . . .	944
10.264.4.4GetSegments() [1/2] . . . . .	944
10.264.4.5GetSegments() [2/2] . . . . .	944
10.264.4.6PrepareWrite() . . . . .	944
10.264.4.7SetNumberOfSegments() . . . . .	945



10.264.4.8SetSegments()	945
10.264.4.9Write()	945
10.264.5Member Data Documentation	945
10.264.5.1Segments	945
10.265dcm::SequenceOfFragments Class Reference	946
10.265.1Detailed Description	947
10.265.2Member Typedef Documentation	948
10.265.2.1ConstIterator	948
10.265.2.2FragmentVector	948
10.265.2.3Iterator	948
10.265.2.4SizeType	948
10.265.3Constructor & Destructor Documentation	948
10.265.3.1SequenceOfFragments()	948
10.265.4Member Function Documentation	948
10.265.4.1AddFragment()	949
10.265.4.2Begin() [1/2]	949
10.265.4.3Begin() [2/2]	949
10.265.4.4Clear()	949
10.265.4.5ComputeByteLength()	949
10.265.4.6ComputeLength()	949
10.265.4.7End() [1/2]	950
10.265.4.8End() [2/2]	950
10.265.4.9GetBuffer()	950
10.265.4.10GetFragBuffer()	950
10.265.4.11GetFragment()	950
10.265.4.12GetLength()	951
10.265.4.13GetNumberOfFragments()	951
10.265.4.14GetTable() [1/2]	951

10.265.4.10SetTable() [2/2]	951
10.265.4.11New()	951
10.265.4.12operator==( )	951
10.265.4.13Print()	952
10.265.4.14Read()	952
10.265.4.20ReadPreValue()	952
10.265.4.21ReadValue()	952
10.265.4.22SetLength()	952
10.265.4.23Write()	953
10.265.4.24WriteBuffer()	953
10.266.0dcm::SequenceOfItems Class Reference	953
10.266.1Detailed Description	955
10.266.2Member Typedef Documentation	956
10.266.2.1ConstIterator	956
10.266.2.2ItemVector	956
10.266.2.3Iterator	956
10.266.2.4SizeType	956
10.266.3Constructor & Destructor Documentation	956
10.266.3.1SequenceOfItems()	957
10.266.4Member Function Documentation	957
10.266.4.1AddItem()	957
10.266.4.2AddNewUndefinedLengthItem()	957
10.266.4.3Begin() [1/2]	957
10.266.4.4Begin() [2/2]	957
10.266.4.5Clear()	958
10.266.4.6ComputeLength()	958
10.266.4.7End() [1/2]	958
10.266.4.8End() [2/2]	958

10.266.4.9FindDataElement()	958
10.266.4.10GetItem() [1/2]	958
10.266.4.10GetItem() [2/2]	959
10.266.4.10GetLength()	959
10.266.4.10GetNumberOfItems()	959
10.266.4.11Empty()	959
10.266.4.11UndefinedLength()	959
10.266.4.11New()	960
10.266.4.10operator=()	960
10.266.4.10operator==()	960
10.266.4.10Print()	960
10.266.4.20Read()	961
10.266.4.21RemoveItemByIndex()	961
10.266.4.22SetLength()	961
10.266.4.23SetLengthToUndefined()	961
10.266.4.23SetNumberOfItems()	962
10.266.4.25Write()	962
10.266.5Member Data Documentation	962
10.266.5.1Items	962
10.266.5.2SequenceLengthField	962
10.267dcm::SerieHelper Class Reference	963
10.267.1Detailed Description	964
10.267.2Member Typedef Documentation	964
10.267.2.1SerieRestrictions	964
10.267.2.2SingleSerieUIDFileSetmap	964
10.267.3Constructor & Destructor Documentation	964
10.267.3.1SerieHelper()	965
10.267.3.2~SerieHelper()	965

10.267.4	Member Function Documentation	965
10.267.4.1	AddFile()	965
10.267.4.2	AddFileName()	965
10.267.4.3	AddRestriction() [1/3]	965
10.267.4.4	AddRestriction() [2/3]	965
10.267.4.5	AddRestriction() [3/3]	966
10.267.4.6	Clear()	966
10.267.4.7	CreateDefaultUniqueSeriesIdentifier()	966
10.267.4.8	CreateUniqueSeriesIdentifier()	966
10.267.4.9	FileNameOrdering()	966
10.267.4.10	GetFirstSingleSerieUIDFileSet()	966
10.267.4.11	GetNextSingleSerieUIDFileSet()	966
10.267.4.12	ImagePositionPatientOrdering()	967
10.267.4.13	OrderFileList()	967
10.267.4.14	SetDirectory()	967
10.267.4.15	SetLoadMode()	967
10.267.4.16	SetUseSeriesDetails()	967
10.267.4.17	UserOrdering()	967
10.267.5	Member Data Documentation	967
10.267.5.1	ItFileSetHt	968
10.267.5.2	SingleSerieUIDFileSetHT	968
10.268	dcm::Series Class Reference	968
10.268.1	Detailed Description	968
10.268.2	Constructor & Destructor Documentation	968
10.268.2.1	Series()	968
10.269	dcm::network::ServiceClassApplicationInformation Class Reference	969
10.269.1	Detailed Description	969
10.269.2	Constructor & Destructor Documentation	969

10.269.2.1ServiceClassApplicationInformation()	. . . . .	969
10.269.3Member Function Documentation	. . . . .	969
10.269.3.1Print()	. . . . .	969
10.269.3.2Read()	. . . . .	969
10.269.3.3SetTuple()	. . . . .	970
10.269.3.4Size()	. . . . .	970
10.269.3.5Write()	. . . . .	970
10.270dcm::ServiceClassUser Class Reference	. . . . .	970
10.270.1Detailed Description	. . . . .	972
10.270.2Constructor & Destructor Documentation	. . . . .	972
10.270.2.1ServiceClassUser()	. . . . .	972
10.270.2.2~ServiceClassUser()	. . . . .	973
10.270.3Member Function Documentation	. . . . .	973
10.270.3.1GetAETitle()	. . . . .	973
10.270.3.2GetCalledAETitle()	. . . . .	973
10.270.3.3GetTimeout()	. . . . .	973
10.270.3.4InitializeConnection()	. . . . .	973
10.270.3.5IsPresentationContextAccepted()	. . . . .	974
10.270.3.6New()	. . . . .	974
10.270.3.7SendEcho()	. . . . .	974
10.270.3.8SendFind()	. . . . .	974
10.270.3.9SendMove() [1/3]	. . . . .	974
10.270.3.10SendMove() [2/3]	. . . . .	975
10.270.3.11SendMove() [3/3]	. . . . .	975
10.270.3.12SendStore() [1/3]	. . . . .	975
10.270.3.13SendStore() [2/3]	. . . . .	975
10.270.3.14SendStore() [3/3]	. . . . .	975
10.270.3.15SetAETitle()	. . . . .	976

10.270.3.1	<del>Set</del> CalledAETitle()	976
10.270.3.1	<del>Set</del> Hostname()	976
10.270.3.1	<del>Set</del> Port()	976
10.270.3.1	<del>Set</del> PortSCP()	977
10.270.3.2	<del>Set</del> PresentationContexts()	977
10.270.3.2	<del>Set</del> Timeout()	977
10.270.3.2	<del>Start</del> Association()	977
10.270.3.2	<del>Stop</del> Association()	978
10.270	dcm::SHA1 Class Reference	978
10.271.1	Detailed Description	978
10.271.2	Constructor & Destructor Documentation	979
10.271.2.1	SHA1()	979
10.271.2.2	~SHA1()	979
10.271.3	Member Function Documentation	979
10.271.3.1	Compute()	979
10.271.3.2	ComputeFile()	979
10.270	dcm::SimpleMemberCommand< T > Class Template Reference	980
10.272.1	Detailed Description	982
10.272.2	Member Typedef Documentation	982
10.272.2.1	Self	982
10.272.2.2	TMemberFunctionPointer	982
10.272.3	Constructor & Destructor Documentation	982
10.272.3.1	SimpleMemberCommand()	982
10.272.3.2	~SimpleMemberCommand()	983
10.272.4	Member Function Documentation	983
10.272.4.1	Execute() [ 1 / 2 ]	983
10.272.4.2	Execute() [ 2 / 2 ]	983
10.272.4.3	New()	983

10.272.4.4SetCallbackFunction()	984
10.272.5Member Data Documentation	984
10.272.5.1m_MemberFunction	984
10.272.5.2m_This	984
10.273dcm::SimpleSubjectWatcher Class Reference	984
10.273.1Detailed Description	985
10.273.2Constructor & Destructor Documentation	985
10.273.2.1SimpleSubjectWatcher()	985
10.273.2.2~SimpleSubjectWatcher()	985
10.273.3Member Function Documentation	985
10.273.3.1EndFilter()	986
10.273.3.2ShowAbort()	986
10.273.3.3ShowAnonymization()	986
10.273.3.4ShowData()	986
10.273.3.5ShowDataSet()	986
10.273.3.6ShowFileName()	986
10.273.3.7ShowIteration()	987
10.273.3.8ShowProgress()	987
10.273.3.9StartFilter()	987
10.273.3.10TestAbortOff()	987
10.273.3.11TestAbortOn()	987
10.274dcm::MrProtocol::Slice Struct Reference	988
10.274.1Member Data Documentation	988
10.274.1.1Normal	988
10.274.1.2Position	988
10.275dcm::MrProtocol::SliceArray Struct Reference	989
10.275.1Member Data Documentation	989
10.275.1.1Slices	989

10.276	dcm::SmartPointer< ObjectType > Class Template Reference . . . . .	990
10.276.1	Detailed Description . . . . .	991
10.276.2	Constructor & Destructor Documentation . . . . .	991
10.276.2.1	SmartPointer() [1/4] . . . . .	992
10.276.2.2	SmartPointer() [2/4] . . . . .	992
10.276.2.3	SmartPointer() [3/4] . . . . .	992
10.276.2.4	SmartPointer() [4/4] . . . . .	992
10.276.2.5	~SmartPointer() . . . . .	992
10.276.3	Member Function Documentation . . . . .	992
10.276.3.1	GetPointer() . . . . .	993
10.276.3.2	operator ObjectType *() . . . . .	993
10.276.3.3	operator*() . . . . .	993
10.276.3.4	operator->() . . . . .	993
10.276.3.5	operator=() [1/3] . . . . .	993
10.276.3.6	operator=() [2/3] . . . . .	994
10.276.3.7	operator=() [3/3] . . . . .	994
10.277	dcm::network::SOPClassExtendedNegociationSub Class Reference . . . . .	994
10.277.1	Detailed Description . . . . .	994
10.277.2	Constructor & Destructor Documentation . . . . .	995
10.277.2.1	SOPClassExtendedNegociationSub() . . . . .	995
10.277.3	Member Function Documentation . . . . .	995
10.277.3.1	Print() . . . . .	995
10.277.3.2	Read() . . . . .	995
10.277.3.3	SetTuple() . . . . .	995
10.277.3.4	Size() . . . . .	995
10.277.3.5	Write() . . . . .	996
10.278	dcm::SOPClassUIDToIOD Class Reference . . . . .	996
10.278.1	Detailed Description . . . . .	996



10.278.2	Member Typedef Documentation . . . . .	996
10.278.2.1	const . . . . .	996
10.278.3	Member Function Documentation . . . . .	997
10.278.3.1	GetIOD() . . . . .	997
10.278.3.2	GetIODFromSOPClassUID() . . . . .	997
10.278.3.3	GetNumberOfSOPClassToIOD() . . . . .	997
10.278.3.4	GetSOPClassUIDFromIOD() . . . . .	997
10.278.3.5	GetSOPClassUIDToIOD() . . . . .	997
10.278.3.6	GetSOPClassUIDToIODs() . . . . .	998
10.279	dcm::Sorter Class Reference . . . . .	998
10.279.1	Detailed Description . . . . .	999
10.279.2	Member Typedef Documentation . . . . .	1000
10.279.2.1	SelectionMap . . . . .	1000
10.279.2.2	SortFunction . . . . .	1000
10.279.3	Constructor & Destructor Documentation . . . . .	1000
10.279.3.1	Sorter() . . . . .	1000
10.279.3.2	~Sorter() . . . . .	1000
10.279.4	Member Function Documentation . . . . .	1000
10.279.4.1	AddSelect() . . . . .	1000
10.279.4.2	GetFileNames() . . . . .	1001
10.279.4.3	Print() . . . . .	1001
10.279.4.4	SetSortFunction() . . . . .	1001
10.279.4.5	SetTagsToRead() . . . . .	1001
10.279.4.6	Sort() . . . . .	1002
10.279.4.7	StableSort() . . . . .	1002
10.279.5	Friends And Related Function Documentation . . . . .	1002
10.279.5.1	operator<< . . . . .	1002
10.279.6	Member Data Documentation . . . . .	1002

10.279.6.1	FileNames	1002
10.279.6.2	Selection	1003
10.279.6.3	SortFunc	1003
10.279.6.4	TagsToRead	1003
10.280	dcm::Spacing Class Reference	1003
10.280.1	Detailed Description	1004
10.280.2	Member Enumeration Documentation	1004
10.280.2.1	SpacingType	1004
10.280.3	Constructor & Destructor Documentation	1005
10.280.3.1	Spacing()	1005
10.280.3.2	~Spacing()	1005
10.280.4	Member Function Documentation	1005
10.280.4.1	ComputePixelAspectRatioFromPixelSpacing()	1005
10.281	dcm::Spectroscopy Class Reference	1005
10.281.1	Detailed Description	1006
10.281.2	Constructor & Destructor Documentation	1006
10.281.2.1	Spectroscopy()	1006
10.282	dcm::SplitMosaicFilter Class Reference	1006
10.282.1	Detailed Description	1007
10.282.2	Constructor & Destructor Documentation	1007
10.282.2.1	SplitMosaicFilter()	1007
10.282.2.2	~SplitMosaicFilter()	1007
10.282.3	Member Function Documentation	1007
10.282.3.1	ComputeMOSAICDimensions()	1007
10.282.3.2	ComputeMOSAICSliceNormal()	1007
10.282.3.3	ComputeMOSAICSlicePosition()	1008
10.282.3.4	GetFile() [1/2]	1008
10.282.3.5	GetFile() [2/2]	1008

10.282.3.6	<a href="#">GetImage() [1/2]</a>	1008
10.282.3.7	<a href="#">GetImage() [2/2]</a>	1008
10.282.3.8	<a href="#">SetFile()</a>	1008
10.282.3.9	<a href="#">SetImage()</a>	1009
10.282.3.10	<a href="#">Split()</a>	1009
10.283	<a href="#">dcm::StartEvent Class Reference</a>	1009
10.284	<a href="#">dcm::static_assert_test&lt; x &gt; Struct Template Reference</a>	1010
10.285	<a href="#">dcm::STATIC_ASSERTION_FAILURE&lt; x &gt; Struct Template Reference</a>	1010
10.286	<a href="#">dcm::STATIC_ASSERTION_FAILURE&lt; true &gt; Struct Template Reference</a>	1011
10.286.1	<a href="#">Member Enumeration Documentation</a>	1011
10.286.1.1	<a href="#">anonymous enum</a>	1011
10.287	<a href="#">dcm::StreamImageReader Class Reference</a>	1011
10.287.1	<a href="#">Detailed Description</a>	1012
10.287.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1012
10.287.2.1	<a href="#">StreamImageReader()</a>	1012
10.287.2.2	<a href="#">~StreamImageReader()</a>	1012
10.287.3	<a href="#">Member Function Documentation</a>	1012
10.287.3.1	<a href="#">CanReadImage()</a>	1012
10.287.3.2	<a href="#">DefinePixelExtent()</a>	1013
10.287.3.3	<a href="#">DefineProperBufferLength()</a>	1013
10.287.3.4	<a href="#">GetDimensionsValueForResolution()</a>	1013
10.287.3.5	<a href="#">GetFile()</a>	1013
10.287.3.6	<a href="#">Read()</a>	1014
10.287.3.7	<a href="#">ReadImageInformation()</a>	1014
10.287.3.8	<a href="#">SetFileName()</a>	1014
10.287.3.9	<a href="#">SetStream()</a>	1015
10.288	<a href="#">dcm::StreamImageWriter Class Reference</a>	1015
10.288.1	<a href="#">Detailed Description</a>	1017

10.288.2	Constructor & Destructor Documentation	1017
10.288.2.1	StreamImageWriter()	1018
10.288.2.2	~StreamImageWriter()	1018
10.288.3	Member Function Documentation	1018
10.288.3.1	CanWriteFile()	1018
10.288.3.2	DefinePixelExtent()	1018
10.288.3.3	DefineProperBufferLength()	1019
10.288.3.4	SetFile()	1019
10.288.3.5	SetFileName()	1019
10.288.3.6	SetStream()	1019
10.288.3.7	Write()	1020
10.288.3.8	WriteImageInformation()	1020
10.288.3.9	WriteImageSubregionRAW()	1020
10.288.3.10	WriteRawHeader()	1021
10.288.4	Member Data Documentation	1021
10.288.4.1	mElementOffsets	1021
10.288.4.2	mElementOffsets1	1021
10.288.4.3	mFile	1021
10.288.4.4	mWriter	1021
10.288.4.5	mXMax	1022
10.288.4.6	mXMin	1022
10.288.4.7	mYMax	1022
10.288.4.8	mYMin	1022
10.288.4.9	mZMax	1022
10.288.4.10	mZMin	1022
10.289	dcm::StrictScanner Class Reference	1023
10.289.1	Detailed Description	1025
10.289.2	Member Typedef Documentation	1025

10.289.2.1ConstIterator . . . . .	1025
10.289.2.2MappingType . . . . .	1026
10.289.2.3TagToValue . . . . .	1026
10.289.2.4TagToValueValueType . . . . .	1026
10.289.2.5ValuesType . . . . .	1026
10.289.3Constructor & Destructor Documentation . . . . .	1026
10.289.3.1StrictScanner() . . . . .	1026
10.289.3.2~StrictScanner() . . . . .	1026
10.289.4Member Function Documentation . . . . .	1026
10.289.4.1AddPrivateTag() . . . . .	1027
10.289.4.2AddSkipTag() . . . . .	1027
10.289.4.3AddTag() . . . . .	1027
10.289.4.4Begin() . . . . .	1027
10.289.4.5ClearSkipTags() . . . . .	1027
10.289.4.6ClearTags() . . . . .	1027
10.289.4.7End() . . . . .	1028
10.289.4.8GetAllFileNamesFromTagToValue() . . . . .	1028
10.289.4.9GetFilenameFromTagToValue() . . . . .	1028
10.289.4.10GetFileNames() . . . . .	1028
10.289.4.11GetKeys() . . . . .	1028
10.289.4.12GetMapping() . . . . .	1028
10.289.4.13GetMappingFromTagToValue() . . . . .	1029
10.289.4.14GetMappings() . . . . .	1029
10.289.4.15GetOrderedValues() . . . . .	1029
10.289.4.16GetValue() . . . . .	1029
10.289.4.17GetValues() [1/2] . . . . .	1029
10.289.4.18GetValues() [2/2] . . . . .	1030
10.289.4.19Key() . . . . .	1030

10.289.4.1New()	1030
10.289.4.2Print()	1030
10.289.4.3ProcessPublicTag()	1031
10.289.4.4Scan()	1031
10.289.5Friends And Related Function Documentation	1031
10.289.5.1operator<<	1031
10.290dcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	1032
10.290.1Detailed Description	1033
10.290.2Member Typedef Documentation	1034
10.290.2.1const_iterator	1034
10.290.2.2const_reference	1034
10.290.2.3const_reverse_iterator	1034
10.290.2.4difference_type	1034
10.290.2.5iterator	1034
10.290.2.6pointer	1035
10.290.2.7reference	1035
10.290.2.8reverse_iterator	1035
10.290.2.9size_type	1035
10.290.2.10value_type	1035
10.290.3Constructor & Destructor Documentation	1035
10.290.3.1String() [1/4]	1035
10.290.3.2String() [2/4]	1036
10.290.3.3String() [3/4]	1036
10.290.3.4String() [4/4]	1036
10.290.4Member Function Documentation	1036
10.290.4.1IsValid()	1036
10.290.4.2operator const char *()	1036
10.290.4.3Trim() [1/2]	1037

10.290.4.4Trim() [2/2]	1037
10.290.4.5Truncate()	1037
10.291dcm::StringFilter Class Reference	1037
10.291.1Detailed Description	1038
10.291.2Constructor & Destructor Documentation	1038
10.291.2.1StringFilter()	1038
10.291.2.2~StringFilter()	1038
10.291.3Member Function Documentation	1038
10.291.3.1ExecuteQuery() [1/2]	1038
10.291.3.2ExecuteQuery() [2/2]	1039
10.291.3.3FromString()	1039
10.291.3.4GetFile() [1/2]	1039
10.291.3.5GetFile() [2/2]	1039
10.291.3.6SetDicts()	1039
10.291.3.7SetFile()	1040
10.291.3.8ToString() [1/2]	1040
10.291.3.9ToString() [2/2]	1040
10.291.3.10bStringPair() [1/3]	1040
10.291.3.11bStringPair() [2/3]	1041
10.291.3.12bStringPair() [3/3]	1041
10.291.3.13UseDictAlways()	1041
10.292dcm::Study Class Reference	1041
10.292.1Detailed Description	1041
10.292.2Constructor & Destructor Documentation	1041
10.292.2.1Study()	1042
10.293dcm::Subject Class Reference	1042
10.293.1Detailed Description	1043
10.293.2Constructor & Destructor Documentation	1043

10.293.2.1Subject()	1043
10.293.2.2~Subject()	1043
10.293.3Member Function Documentation	1043
10.293.3.1AddObserver() [1/2]	1044
10.293.3.2AddObserver() [2/2]	1044
10.293.3.3GetCommand()	1044
10.293.3.4HasObserver()	1044
10.293.3.5InvokeEvent() [1/2]	1044
10.293.3.6InvokeEvent() [2/2]	1045
10.293.3.7RemoveAllObservers()	1045
10.293.3.8RemoveObserver()	1045
10.294dcm::Surface Class Reference	1045
10.294.1Detailed Description	1048
10.294.2Member Enumeration Documentation	1048
10.294.2.1STATES	1048
10.294.2.2VIEWType	1048
10.294.3Constructor & Destructor Documentation	1049
10.294.3.1Surface()	1049
10.294.3.2~Surface()	1049
10.294.4Member Function Documentation	1049
10.294.4.1GetAlgorithmFamily() [1/2]	1049
10.294.4.2GetAlgorithmFamily() [2/2]	1049
10.294.4.3GetAlgorithmName()	1049
10.294.4.4GetAlgorithmVersion()	1050
10.294.4.5GetAxisOfRotation()	1050
10.294.4.6GetCenterOfRotation()	1050
10.294.4.7GetFiniteVolume()	1050
10.294.4.8GetManifold()	1050



10.294.4.9	GetMaximumPointDistance()	1050
10.294.4.10	GetMeanPointDistance()	1051
10.294.4.10	GetMeshPrimitive() [1/2]	1051
10.294.4.10	GetMeshPrimitive() [2/2]	1051
10.294.4.10	GetNumberOfSurfacePoints()	1051
10.294.4.10	GetNumberOfVectors()	1051
10.294.4.10	GetPointCoordinatesData() [1/2]	1051
10.294.4.10	GetPointCoordinatesData() [2/2]	1051
10.294.4.10	GetPointPositionAccuracy()	1052
10.294.4.10	GetPointsBoundingBoxCoordinates()	1052
10.294.4.10	GetProcessingAlgorithm() [1/2]	1052
10.294.4.20	GetProcessingAlgorithm() [2/2]	1052
10.294.4.20	GetRecommendedDisplayCIELabValue() [1/2]	1052
10.294.4.20	GetRecommendedDisplayCIELabValue() [2/2]	1052
10.294.4.20	GetRecommendedDisplayGrayscaleValue()	1053
10.294.4.20	GetRecommendedPresentationOpacity()	1053
10.294.4.20	GetRecommendedPresentationType()	1053
10.294.4.20	GetSTATES()	1053
10.294.4.20	GetSTATESString()	1053
10.294.4.20	GetSurfaceComments()	1053
10.294.4.20	GetSurfaceNumber()	1053
10.294.4.30	GetSurfaceProcessing()	1054
10.294.4.30	GetSurfaceProcessingDescription()	1054
10.294.4.30	GetSurfaceProcessingRatio()	1054
10.294.4.30	GetVectorAccuracy()	1054
10.294.4.30	GetVectorCoordinateData() [1/2]	1054
10.294.4.30	GetVectorCoordinateData() [2/2]	1054
10.294.4.30	GetVectorDimensionality()	1054

10.294.4.37	SetVIEWType()	1055
10.294.4.38	SetVIEWTypeString()	1055
10.294.4.39	SetAlgorithmFamily()	1055
10.294.4.40	SetAlgorithmName()	1055
10.294.4.41	SetAlgorithmVersion()	1055
10.294.4.42	SetAxisOfRotation()	1055
10.294.4.43	SetCenterOfRotation()	1056
10.294.4.44	SetFiniteVolume()	1056
10.294.4.45	SetManifold()	1056
10.294.4.46	SetMaximumPointDistance()	1056
10.294.4.47	SetMeanPointDistance()	1056
10.294.4.48	SetMeshPrimitive()	1056
10.294.4.49	SetNumberOfSurfacePoints()	1057
10.294.4.50	SetNumberOfVectors()	1057
10.294.4.51	SetPointCoordinatesData()	1057
10.294.4.52	SetPointPositionAccuracy()	1057
10.294.4.53	SetPointsBoundingBoxCoordinates()	1057
10.294.4.54	SetProcessingAlgorithm()	1057
10.294.4.55	SetRecommendedDisplayCIELabValue() [1/3]	1058
10.294.4.56	SetRecommendedDisplayCIELabValue() [2/3]	1058
10.294.4.57	SetRecommendedDisplayCIELabValue() [3/3]	1058
10.294.4.58	SetRecommendedDisplayGrayscaleValue()	1058
10.294.4.59	SetRecommendedPresentationOpacity()	1058
10.294.4.60	SetRecommendedPresentationType()	1058
10.294.4.61	SetSurfaceComments()	1059
10.294.4.62	SetSurfaceNumber()	1059
10.294.4.63	SetSurfaceProcessing()	1059
10.294.4.64	SetSurfaceProcessingDescription()	1059

10.294.4.65	SetSurfaceProcessingRatio()	1059
10.294.4.66	SetVectorAccuracy()	1059
10.294.4.67	SetVectorCoordinateData()	1060
10.294.4.68	SetVectorDimensionality()	1060
10.295	dcm::SurfaceHelper Class Reference	1060
10.295.1	Detailed Description	1061
10.295.2	Member Typedef Documentation	1061
10.295.2.1	ColorArray	1061
10.295.3	Member Function Documentation	1061
10.295.3.1	RecommendedDisplayCIELabToRGB() [1/2]	1061
10.295.3.2	RecommendedDisplayCIELabToRGB() [2/2]	1062
10.295.3.3	RGBToRecommendedDisplayCIELab()	1062
10.295.3.4	RGBToRecommendedDisplayGrayscale()	1063
10.296	dcm::SurfaceReader Class Reference	1063
10.296.1	Detailed Description	1065
10.296.2	Constructor & Destructor Documentation	1065
10.296.2.1	SurfaceReader()	1065
10.296.2.2	~SurfaceReader()	1065
10.296.3	Member Function Documentation	1065
10.296.3.1	GetNumberOfSurfaces()	1066
10.296.3.2	Read()	1066
10.296.3.3	ReadPointMacro()	1066
10.296.3.4	ReadSurface()	1066
10.296.3.5	ReadSurfaces()	1066
10.297	dcm::SurfaceWriter Class Reference	1067
10.297.1	Detailed Description	1068
10.297.2	Constructor & Destructor Documentation	1068
10.297.2.1	SurfaceWriter()	1068

10.297.2.2~SurfaceWriter()	1068
10.297.3Member Function Documentation	1068
10.297.3.1ComputeNumberOfSurfaces()	1068
10.297.3.2GetNumberOfSurfaces()	1069
10.297.3.3PrepareWrite()	1069
10.297.3.4PrepareWritePointMacro()	1069
10.297.3.5SetNumberOfSurfaces()	1069
10.297.3.6Write()	1069
10.297.4Member Data Documentation	1069
10.297.4.1NumberOfSurfaces	1070
10.298dcm::SwapCode Class Reference	1070
10.298.1Detailed Description	1071
10.298.2Member Enumeration Documentation	1071
10.298.2.1SwapCodeType	1071
10.298.3Constructor & Destructor Documentation	1071
10.298.3.1SwapCode()	1071
10.298.4Member Function Documentation	1071
10.298.4.1GetIndex()	1072
10.298.4.2GetSwapCodeString()	1072
10.298.4.3operator SwapCode::SwapCodeType()	1072
10.298.5Friends And Related Function Documentation	1072
10.298.5.1operator<<	1072
10.299dcm::SwapperDoOp Class Reference	1072
10.299.1Member Function Documentation	1073
10.299.1.1Swap()	1073
10.299.1.2SwapArray()	1073
10.300dcm::SwapperNoOp Class Reference	1073
10.300.1Detailed Description	1073

10.300.2	Member Function Documentation	1073
10.300.2.1	Swap()	1074
10.300.2.2	SwapArray()	1074
10.300.4	dcm::System Class Reference	1074
10.301.1	Detailed Description	1075
10.301.2	Member Function Documentation	1075
10.301.2.1	DeleteDirectory()	1075
10.301.2.2	EncodeBytes()	1076
10.301.2.3	FileExists()	1076
10.301.2.4	FileIsDirectory()	1076
10.301.2.5	FileIsSymlink()	1076
10.301.2.6	FileSize()	1077
10.301.2.7	FileTime()	1077
10.301.2.8	FormatDateTime()	1077
10.301.2.9	GetCurrentDateTime()	1077
10.301.2.10	GetCurrentModuleFileName()	1078
10.301.2.11	GetCurrentProcessFileName()	1078
10.301.2.12	GetCurrentResourcesDirectory()	1078
10.301.2.13	GetCurrentCWD()	1078
10.301.2.14	GetHostName()	1078
10.301.2.15	GetLastSystemError()	1078
10.301.2.16	GetLocaleCharset()	1079
10.301.2.17	GetPermissions()	1079
10.301.2.18	GetTimezoneOffsetFromUTC()	1079
10.301.2.19	MakeDirectory()	1079
10.301.2.20	ParseDateTime() [1/2]	1079
10.301.2.21	ParseDateTime() [2/2]	1080
10.301.2.22	RemoveFile()	1080

10.301.2.2	SetPermissions()	1080
10.301.2.2	StrCaseCmp()	1080
10.301.2.2	StrNCaseCmp()	1081
10.301.2.2	StrSep()	1081
10.301.2.2	StrTokR()	1081
10.302	dcm::Table Class Reference	1081
10.302.1	Detailed Description	1082
10.302.2	Member Typedef Documentation	1082
10.302.2.1	MapTableEntry	1082
10.302.3	Constructor & Destructor Documentation	1082
10.302.3.1	Table()	1082
10.302.3.2	~Table()	1082
10.302.4	Member Function Documentation	1083
10.302.4.1	GetTableEntry()	1083
10.302.4.2	InsertEntry()	1083
10.302.5	Friends And Related Function Documentation	1083
10.302.5.1	operator<<	1083
10.303	dcm::TableEntry Class Reference	1083
10.303.1	Detailed Description	1084
10.303.2	Constructor & Destructor Documentation	1084
10.303.2.1	TableEntry()	1084
10.303.2.2	~TableEntry()	1084
10.304	dcm::TableReader Class Reference	1084
10.304.1	Detailed Description	1085
10.304.2	Constructor & Destructor Documentation	1085
10.304.2.1	TableReader()	1085
10.304.2.2	~TableReader()	1085
10.304.3	Member Function Documentation	1086

10.304.3.1CharacterDataHandler()	1086
10.304.3.2EndElement()	1086
10.304.3.3GetDefs()	1086
10.304.3.4GetFilename()	1086
10.304.3.5HandleIOD()	1086
10.304.3.6HandleIODEntry()	1087
10.304.3.7HandleMacro()	1087
10.304.3.8HandleMacroEntry()	1087
10.304.3.9HandleMacroEntryDescription()	1087
10.304.3.10HandleModule()	1087
10.304.3.11HandleModuleEntry()	1087
10.304.3.12HandleModuleEntryDescription()	1088
10.304.3.13HandleModuleInclude()	1088
10.304.3.14Read()	1088
10.304.3.15SetFilename()	1088
10.304.3.16StartElement()	1088
10.305dcm::network::TableRow Class Reference	1089
10.305.1Constructor & Destructor Documentation	1089
10.305.1.1TableRow()	1089
10.305.1.2~TableRow()	1090
10.305.2Member Data Documentation	1090
10.305.2.1transitions	1090
10.306dcm::Tag Class Reference	1090
10.306.1Detailed Description	1092
10.306.2Constructor & Destructor Documentation	1092
10.306.2.1Tag() [1/3]	1093
10.306.2.2Tag() [2/3]	1093
10.306.2.3Tag() [3/3]	1093

10.306.3	Member Function Documentation	1093
10.306.3.1	GetElement()	1093
10.306.3.2	GetElementTag()	1094
10.306.3.3	GetGroup()	1094
10.306.3.4	GetLength()	1094
10.306.3.5	GetPrivateCreator()	1094
10.306.3.6	IsGroupLength()	1095
10.306.3.7	IsGroupXX()	1095
10.306.3.8	IsIllegal()	1095
10.306.3.9	IsPrivate()	1095
10.306.3.10	IsPrivateCreator()	1096
10.306.3.11	IsPublic()	1096
10.306.3.12	Operator"!="()	1096
10.306.3.13	Operator"<()"	1096
10.306.3.14	Operator"<=()"	1096
10.306.3.15	Operator"=()"	1097
10.306.3.16	Operator"==()"	1097
10.306.3.17	Operator[]"() [1/2]	1097
10.306.3.18	Operator[]"() [2/2]	1097
10.306.3.19	PrintAsContinuousString()	1097
10.306.3.20	PrintAsContinuousUpperCaseString()	1098
10.306.3.21	PrintAsPipeSeparatedString()	1098
10.306.3.22	Read()	1098
10.306.3.23	ReadFromCommaSeparatedString()	1098
10.306.3.24	ReadFromContinuousString()	1098
10.306.3.25	ReadFromPipeSeparatedString()	1099
10.306.3.26	SetElement()	1099
10.306.3.27	SetElementTag() [1/2]	1099



10.306.3.28	SetElementTag() [2/2]	1099
10.306.3.29	SetGroup()	1100
10.306.3.30	SetPrivateCreator()	1100
10.306.3.31	Write()	1100
10.306.4	Friends And Related Function Documentation	1100
10.306.4.1	operator<<	1100
10.306.4.2	operator>>	1101
10.306.5	Member Data Documentation	1101
10.306.5.1	bytes	1101
10.306.5.2	tag	1101
10.306.5.3	tags	1101
10.307	dcm::TagPath Class Reference	1101
10.307.1	Detailed Description	1102
10.307.2	Constructor & Destructor Documentation	1102
10.307.2.1	TagPath()	1102
10.307.2.2	~TagPath()	1102
10.307.3	Member Function Documentation	1102
10.307.3.1	ConstructFromString()	1103
10.307.3.2	ConstructFromTagList()	1103
10.307.3.3	IsValid()	1103
10.307.3.4	Print()	1103
10.307.3.5	Push() [1/2]	1103
10.307.3.6	Push() [2/2]	1104
10.308	dcm::Testing Class Reference	1104
10.308.1	Detailed Description	1105
10.308.2	Member Typedef Documentation	1105
10.308.2.1	MD5DataImagesType	1105
10.308.2.2	MediaStorageDataFilesType	1105

10.308.3	Constructor & Destructor Documentation . . . . .	1106
10.308.3.1	Testing() . . . . .	1106
10.308.3.2	~Testing() . . . . .	1106
10.308.4	Member Function Documentation . . . . .	1106
10.308.4.1	ComputeFileMD5() . . . . .	1106
10.308.4.2	ComputeMD5() . . . . .	1106
10.308.4.3	GetDataExtraRoot() . . . . .	1107
10.308.4.4	GetDataRoot() . . . . .	1107
10.308.4.5	GetFileName() . . . . .	1107
10.308.4.6	GetFileNames() . . . . .	1107
10.308.4.7	GetLossyFlagFromFile() . . . . .	1108
10.308.4.8	GetMD5DataImage() . . . . .	1108
10.308.4.9	GetMD5DataImages() . . . . .	1108
10.308.4.10	GetMD5FromBrokenFile() . . . . .	1108
10.308.4.10	GetMD5FromFile() . . . . .	1108
10.308.4.10	GetMediaStorageDataFile() . . . . .	1108
10.308.4.10	GetMediaStorageDataFiles() . . . . .	1109
10.308.4.10	GetMediaStorageFromFile() . . . . .	1109
10.308.4.10	GetNumberOfFileNames() . . . . .	1109
10.308.4.10	GetNumberOfMD5DataImages() . . . . .	1109
10.308.4.10	GetNumberOfMediaStorageDataFiles() . . . . .	1109
10.308.4.10	GetPixelSpacingDataRoot() . . . . .	1109
10.308.4.10	GetSelectedPrivateGroupOffsetFromFile() . . . . .	1110
10.308.4.20	GetSelectedTagsOffsetFromFile() . . . . .	1110
10.308.4.20	GetSourceDirectory() . . . . .	1110
10.308.4.20	GetStreamOffsetFromFile() . . . . .	1110
10.308.4.20	GetTempDirectory() . . . . .	1110
10.308.4.20	GetTempDirectoryW() . . . . .	1111

10.308.4.25	SetTempFilename()	1111
10.308.4.26	GetTempFilenameW()	1111
10.308.4.27	Print()	1111
10.309	dcm::Trace Class Reference	1112
10.309.1	Detailed Description	1113
10.309.2	Constructor & Destructor Documentation	1113
10.309.2.1	Trace()	1113
10.309.2.2	~Trace()	1113
10.309.3	Member Function Documentation	1113
10.309.3.1	DebugOff()	1113
10.309.3.2	DebugOn()	1114
10.309.3.3	ErrorOff()	1114
10.309.3.4	ErrorOn()	1114
10.309.3.5	GetDebugFlag()	1114
10.309.3.6	GetDebugStream()	1114
10.309.3.7	GetErrorFlag()	1114
10.309.3.8	GetErrorStream()	1115
10.309.3.9	GetStream()	1115
10.309.3.10	GetWarningFlag()	1115
10.309.3.10	GetWarningStream()	1115
10.309.3.12	SetDebug()	1115
10.309.3.13	SetDebugStream()	1115
10.309.3.13	SetError()	1116
10.309.3.15	SetErrorStream()	1116
10.309.3.15	SetStream()	1116
10.309.3.15	SetStreamToFile()	1116
10.309.3.18	SetWarning()	1116
10.309.3.19	SetWarningStream()	1117

10.309.3.20	WarningOff()	1117
10.309.3.21	WarningOn()	1117
10.310	dcm::TransferSyntax Class Reference	1117
10.310.1	Detailed Description	1119
10.310.2	Member Enumeration Documentation	1119
10.310.2.1	NegotiatedType	1119
10.310.2.2	TSType	1120
10.310.3	Constructor & Destructor Documentation	1120
10.310.3.1	TransferSyntax()	1121
10.310.4	Member Function Documentation	1121
10.310.4.1	CanStoreLossy()	1121
10.310.4.2	GetNegotiatedType()	1121
10.310.4.3	GetString()	1121
10.310.4.4	GetSwapCode()	1121
10.310.4.5	GetTSSString()	1122
10.310.4.6	GetTSType()	1122
10.310.4.7	IsEncapsulated()	1122
10.310.4.8	IsEncoded()	1122
10.310.4.9	IsExplicit()	1122
10.310.4.10	Implicit()	1123
10.310.4.11	IsLossless()	1123
10.310.4.12	IsLossy()	1123
10.310.4.13	IsValid()	1123
10.310.4.14	operator TSType()	1123
10.310.5	Friends And Related Function Documentation	1123
10.310.5.1	operator<<	1123
10.311	dcm::network::TransferSyntaxSub Class Reference	1124
10.311.1	Detailed Description	1124

10.311.2	Constructor & Destructor Documentation	1124
10.311.2.1	TransferSyntaxSub()	1124
10.311.3	Member Function Documentation	1124
10.311.3.1	GetName()	1125
10.311.3.2	operator==( )	1125
10.311.3.3	Print()	1125
10.311.3.4	Read()	1125
10.311.3.5	SetName()	1125
10.311.3.6	SetNameFromUID()	1125
10.311.3.7	Size()	1126
10.311.3.8	Write()	1126
10.312	dcm::network::Transition Struct Reference	1126
10.312.1	Constructor & Destructor Documentation	1127
10.312.1.1	Transition() [1/2]	1127
10.312.1.2	~Transition()	1127
10.312.1.3	Transition() [2/2]	1127
10.312.2	Member Function Documentation	1127
10.312.2.1	MakeNew()	1128
10.312.3	Member Data Documentation	1128
10.312.3.1	mAction	1128
10.312.3.2	mEnd	1128
10.313	dcm::Type Class Reference	1128
10.313.1	Detailed Description	1129
10.313.2	Member Enumeration Documentation	1129
10.313.2.1	TypeType	1129
10.313.3	Constructor & Destructor Documentation	1130
10.313.3.1	Type()	1130
10.313.4	Member Function Documentation	1130

10.313.4.1	GetTypeString()	1130
10.313.4.2	GetTypeType()	1130
10.313.4.3	operator TypeType()	1130
10.313.5	Friends And Related Function Documentation	1130
10.313.5.1	operator<<	1131
10.314	dcm::UI Struct Reference	1131
10.314.1	Friends And Related Function Documentation	1131
10.314.1.1	operator<<	1131
10.314.2	Member Data Documentation	1131
10.314.2.1	Internal	1132
10.315	dcm::UIDGenerator Class Reference	1132
10.315.1	Detailed Description	1132
10.315.2	Constructor & Destructor Documentation	1133
10.315.2.1	UIDGenerator()	1133
10.315.3	Member Function Documentation	1133
10.315.3.1	Generate()	1133
10.315.3.2	GenerateUUID()	1133
10.315.3.3	GetGDCMUID()	1133
10.315.3.4	GetRoot()	1134
10.315.3.5	IsValid()	1134
10.315.3.6	SetRoot()	1134
10.316	dcm::UIDs Class Reference	1134
10.316.1	Detailed Description	1145
10.316.2	Member Typedef Documentation	1145
10.316.2.1	TransferSyntaxStringsType	1145
10.316.3	Member Enumeration Documentation	1145
10.316.3.1	TSName	1145
10.316.3.2	TSType	1152

10.316.4	Member Function Documentation	1158
10.316.4.1	GetName()	1158
10.316.4.2	GetNumberOfTransferSyntaxStrings()	1159
10.316.4.3	GetString()	1159
10.316.4.4	GetTransferSyntaxString()	1159
10.316.4.5	GetTransferSyntaxStrings()	1159
10.316.4.6	GetUIDName()	1159
10.316.4.7	GetUIDString()	1159
10.316.4.8	operator TType()	1160
10.316.4.9	SetFromUID()	1160
10.317	dcm::network::ULAction Class Reference	1160
10.317.1	Detailed Description	1162
10.317.2	Constructor & Destructor Documentation	1162
10.317.2.1	ULAction()	1162
10.317.2.2	~ULAction()	1163
10.317.3	Member Function Documentation	1163
10.317.3.1	PerformAction()	1163
10.318	dcm::network::ULActionAA1 Class Reference	1163
10.318.1	Member Function Documentation	1164
10.318.1.1	PerformAction()	1164
10.319	dcm::network::ULActionAA2 Class Reference	1165
10.319.1	Member Function Documentation	1165
10.319.1.1	PerformAction()	1166
10.320	dcm::network::ULActionAA3 Class Reference	1166
10.320.1	Member Function Documentation	1167
10.320.1.1	PerformAction()	1167
10.321	dcm::network::ULActionAA4 Class Reference	1167
10.321.1	Member Function Documentation	1168

10.321.1.1PerformAction()	1168
10.322dcm::network::ULActionAA5 Class Reference	1169
10.322.1Member Function Documentation	1169
10.322.1.1PerformAction()	1170
10.323dcm::network::ULActionAA6 Class Reference	1170
10.323.1Member Function Documentation	1171
10.323.1.1PerformAction()	1171
10.324dcm::network::ULActionAA7 Class Reference	1171
10.324.1Member Function Documentation	1172
10.324.1.1PerformAction()	1172
10.325dcm::network::ULActionAA8 Class Reference	1173
10.325.1Member Function Documentation	1173
10.325.1.1PerformAction()	1174
10.326dcm::network::ULActionAE1 Class Reference	1174
10.326.1Member Function Documentation	1175
10.326.1.1PerformAction()	1175
10.327dcm::network::ULActionAE2 Class Reference	1175
10.327.1Member Function Documentation	1176
10.327.1.1PerformAction()	1176
10.328dcm::network::ULActionAE3 Class Reference	1177
10.328.1Member Function Documentation	1177
10.328.1.1PerformAction()	1178
10.329dcm::network::ULActionAE4 Class Reference	1178
10.329.1Member Function Documentation	1179
10.329.1.1PerformAction()	1179
10.330dcm::network::ULActionAE5 Class Reference	1179
10.330.1Member Function Documentation	1180
10.330.1.1PerformAction()	1180



10.330	dcm::network::ULActionAE6 Class Reference . . . . .	1181
10.331	Member Function Documentation . . . . .	1181
10.331.1	1.PerformAction() . . . . .	1182
10.330	dcm::network::ULActionAE7 Class Reference . . . . .	1182
10.332	Member Function Documentation . . . . .	1183
10.332.1	1.PerformAction() . . . . .	1183
10.330	dcm::network::ULActionAE8 Class Reference . . . . .	1183
10.333	Member Function Documentation . . . . .	1184
10.333.1	1.PerformAction() . . . . .	1184
10.334	dcm::network::ULActionAR1 Class Reference . . . . .	1185
10.334	Member Function Documentation . . . . .	1185
10.334.1	1.PerformAction() . . . . .	1186
10.335	dcm::network::ULActionAR10 Class Reference . . . . .	1186
10.335	Member Function Documentation . . . . .	1187
10.335.1	1.PerformAction() . . . . .	1187
10.336	dcm::network::ULActionAR2 Class Reference . . . . .	1187
10.336	Member Function Documentation . . . . .	1188
10.336.1	1.PerformAction() . . . . .	1188
10.337	dcm::network::ULActionAR3 Class Reference . . . . .	1189
10.337	Member Function Documentation . . . . .	1189
10.337.1	1.PerformAction() . . . . .	1190
10.338	dcm::network::ULActionAR4 Class Reference . . . . .	1190
10.338	Member Function Documentation . . . . .	1191
10.338.1	1.PerformAction() . . . . .	1191
10.339	dcm::network::ULActionAR5 Class Reference . . . . .	1191
10.339	Member Function Documentation . . . . .	1192
10.339.1	1.PerformAction() . . . . .	1192
10.340	dcm::network::ULActionAR6 Class Reference . . . . .	1193

10.340.1	Member Function Documentation	1193
10.340.1.1	PerformAction()	1194
10.340	dcm::network::ULActionAR7 Class Reference	1194
10.341.1	Member Function Documentation	1195
10.341.1.1	PerformAction()	1195
10.340	dcm::network::ULActionAR8 Class Reference	1195
10.342.1	Member Function Documentation	1196
10.342.1.1	PerformAction()	1196
10.340	dcm::network::ULActionAR9 Class Reference	1197
10.343.1	Member Function Documentation	1197
10.343.1.1	PerformAction()	1198
10.340	dcm::network::ULActionDT1 Class Reference	1198
10.344.1	Member Function Documentation	1199
10.344.1.1	PerformAction()	1199
10.340	dcm::network::ULActionDT2 Class Reference	1199
10.345.1	Member Function Documentation	1200
10.345.1.1	PerformAction()	1200
10.340	dcm::network::ULBasicCallback Class Reference	1201
10.346.1	Detailed Description	1202
10.346.2	Constructor & Destructor Documentation	1202
10.346.2.1	ULBasicCallback()	1202
10.346.2.2	~ULBasicCallback()	1202
10.346.3	Member Function Documentation	1202
10.346.3.1	GetDataSets()	1202
10.346.3.2	GetResponses()	1202
10.346.3.3	HandleDataSet()	1203
10.346.3.4	HandleResponse()	1203
10.340	dcm::network::ULConnection Class Reference	1203

10.347.1 Detailed Description . . . . .	1204
10.347.2 Constructor & Destructor Documentation . . . . .	1204
10.347.2.1 ULCConnection() . . . . .	1204
10.347.2.2 ~ULConnection() . . . . .	1204
10.347.3 Member Function Documentation . . . . .	1204
10.347.3.1 AddAcceptedPresentationContext() . . . . .	1205
10.347.3.2 FindContext() . . . . .	1205
10.347.3.3 GetAcceptedPresentationContexts() [1/2] . . . . .	1205
10.347.3.4 GetAcceptedPresentationContexts() [2/2] . . . . .	1205
10.347.3.5 GetConnectionInfo() . . . . .	1205
10.347.3.6 GetMaxPDUSize() . . . . .	1205
10.347.3.7 GetPresentationContextACByID() . . . . .	1205
10.347.3.8 GetPresentationContextIDFromPresentationContext() . . . . .	1206
10.347.3.9 GetPresentationContextRQByID() . . . . .	1206
10.347.3.10 GetPresentationContexts() . . . . .	1206
10.347.3.11 GetProtocol() . . . . .	1206
10.347.3.12 GetState() . . . . .	1206
10.347.3.13 GetTimer() . . . . .	1206
10.347.3.14 InitializeConnection() . . . . .	1207
10.347.3.15 InitializeIncomingConnection() . . . . .	1207
10.347.3.16 SetMaxPDUSize() . . . . .	1207
10.347.3.17 SetPresentationContexts() [1/2] . . . . .	1207
10.347.3.18 SetPresentationContexts() [2/2] . . . . .	1207
10.347.3.19 SetState() . . . . .	1207
10.347.3.20 StopProtocol() . . . . .	1208
10.347.4 Friends And Related Function Documentation . . . . .	1208
10.347.4.1 ULAActionAE6 . . . . .	1208
10.347.4.2 ULCConnectionManager . . . . .	1208

10.348	dcm::network::ULConnectionCallback Class Reference . . . . .	1208
10.348.1	Detailed Description . . . . .	1209
10.348.2	Constructor & Destructor Documentation . . . . .	1209
10.348.2.1	ULConnectionCallback() . . . . .	1209
10.348.2.2	~ULConnectionCallback() . . . . .	1209
10.348.3	Member Function Documentation . . . . .	1210
10.348.3.1	DataSetHandled() . . . . .	1210
10.348.3.2	DataSetHandles() . . . . .	1210
10.348.3.3	HandleDataSet() . . . . .	1210
10.348.3.4	HandleResponse() . . . . .	1210
10.348.3.5	ResetHandledDataSet() . . . . .	1210
10.348.3.6	SetImplicitFlag() . . . . .	1210
10.348.4	Member Data Documentation . . . . .	1211
10.348.4.1	Implicit . . . . .	1211
10.349	dcm::network::ULConnectionInfo Class Reference . . . . .	1211
10.349.1	Detailed Description . . . . .	1211
10.349.2	Constructor & Destructor Documentation . . . . .	1211
10.349.2.1	ULConnectionInfo() . . . . .	1212
10.349.3	Member Function Documentation . . . . .	1212
10.349.3.1	GetCalledAETitle() . . . . .	1212
10.349.3.2	GetCalledComputerName() . . . . .	1212
10.349.3.3	GetCalledIPAddress() . . . . .	1212
10.349.3.4	GetCalledIPPort() . . . . .	1212
10.349.3.5	GetCallingAETitle() . . . . .	1212
10.349.3.6	GetMaxPDULength() . . . . .	1212
10.349.3.7	Initialize() . . . . .	1213
10.349.3.8	SetMaxPDULength() . . . . .	1213
10.350	dcm::network::ULConnectionManager Class Reference . . . . .	1213

10.350.1	Detailed Description	1215
10.350.2	Constructor & Destructor Documentation	1215
10.350.2.1	ULConnectionManager() [1/2]	1215
10.350.2.2	ULConnectionManager() [2/2]	1215
10.350.2.3	~ULConnectionManager()	1215
10.350.3	Member Function Documentation	1216
10.350.3.1	BreakConnection()	1216
10.350.3.2	BreakConnectionNow()	1216
10.350.3.3	EstablishConnection()	1216
10.350.3.4	EstablishConnectionMove()	1216
10.350.3.5	RunEventLoop()	1217
10.350.3.6	RunMoveEventLoop()	1217
10.350.3.7	SendEcho()	1217
10.350.3.8	SendFind() [1/2]	1217
10.350.3.9	SendFind() [2/2]	1217
10.350.3.10	SendMove() [1/2]	1217
10.350.3.11	SendMove() [2/2]	1218
10.350.3.12	SendNAction() [1/2]	1218
10.350.3.13	SendNAction() [2/2]	1218
10.350.3.14	SendNCreate() [1/2]	1218
10.350.3.15	SendNCreate() [2/2]	1218
10.350.3.16	SendNDelete() [1/2]	1218
10.350.3.17	SendNDelete() [2/2]	1219
10.350.3.18	SendNEventReport() [1/2]	1219
10.350.3.19	SendNEventReport() [2/2]	1219
10.350.3.20	SendNGet() [1/2]	1219
10.350.3.21	SendNGet() [2/2]	1219
10.350.3.22	SendNSet() [1/2]	1219

10.350.3.2SendNSet() [2/2] . . . . .	1220
10.350.3.2SendStore() [1/2] . . . . .	1220
10.350.3.2SendStore() [2/2] . . . . .	1220
10.350.4Member Data Documentation . . . . .	1220
10.350.4.1mConnection . . . . .	1220
10.350.4.2mSecondaryConnection . . . . .	1220
10.350.4.3mTransitions . . . . .	1221
10.351dcm::network::ULEvent Class Reference . . . . .	1221
10.351.1Detailed Description . . . . .	1221
10.351.2Constructor & Destructor Documentation . . . . .	1221
10.351.2.1ULEvent() [1/2] . . . . .	1222
10.351.2.2ULEvent() [2/2] . . . . .	1222
10.351.2.3~ULEvent() . . . . .	1222
10.351.3Member Function Documentation . . . . .	1222
10.351.3.1GetDataSetPos() . . . . .	1222
10.351.3.2GetEvent() . . . . .	1222
10.351.3.3GetIStream() . . . . .	1222
10.351.3.4GetPDUs() . . . . .	1223
10.351.3.5SetEvent() . . . . .	1223
10.351.3.6SetPDU() . . . . .	1223
10.352dcm::network::ULTransitionTable Class Reference . . . . .	1223
10.352.1Detailed Description . . . . .	1223
10.352.2Constructor & Destructor Documentation . . . . .	1224
10.352.2.1ULTransitionTable() . . . . .	1224
10.352.3Member Function Documentation . . . . .	1224
10.352.3.1HandleEvent() . . . . .	1224
10.352.3.2PrintTable() . . . . .	1224
10.353dcm::network::ULWritingCallback Class Reference . . . . .	1225

10.353. Constructor & Destructor Documentation . . . . .	1226
10.353.1.1ULWritingCallback() . . . . .	1226
10.353.1.2~ULWritingCallback() . . . . .	1226
10.353.2 Member Function Documentation . . . . .	1226
10.353.2.1HandleDataSet() . . . . .	1226
10.353.2.2HandleResponse() . . . . .	1226
10.353.2.3SetDirectory() . . . . .	1227
10.354 dcm::UNExplicitDataElement Class Reference . . . . .	1227
10.354.1 Detailed Description . . . . .	1228
10.354.2 Member Function Documentation . . . . .	1228
10.354.2.1GetLength() . . . . .	1229
10.354.2.2Read() . . . . .	1229
10.354.2.3ReadPreValue() . . . . .	1229
10.354.2.4ReadValue() . . . . .	1229
10.354.2.5ReadWithLength() . . . . .	1229
10.355 dcm::UNExplicitImplicitDataElement Class Reference . . . . .	1230
10.355.1 Detailed Description . . . . .	1231
10.355.2 Member Function Documentation . . . . .	1231
10.355.2.1GetLength() . . . . .	1231
10.355.2.2Read() . . . . .	1231
10.355.2.3ReadPreValue() . . . . .	1231
10.355.2.4ReadValue() . . . . .	1232
10.356 dcm::Unpacker12Bits Class Reference . . . . .	1232
10.356.1 Detailed Description . . . . .	1232
10.356.2 Member Function Documentation . . . . .	1232
10.356.2.1Pack() . . . . .	1233
10.356.2.2Unpack() . . . . .	1233
10.357 dcm::Usage Class Reference . . . . .	1233

10.357.1	Detailed Description	1234
10.357.2	Member Enumeration Documentation	1234
10.357.2.1	UsageType	1234
10.357.3	Constructor & Destructor Documentation	1235
10.357.3.1	Usage()	1235
10.357.4	Member Function Documentation	1235
10.357.4.1	GetUsageString()	1235
10.357.4.2	GetUsageType()	1235
10.357.4.3	operator UsageType()	1235
10.357.5	Friends And Related Function Documentation	1235
10.357.5.1	operator<<	1236
10.358	dcm::UserEvent Class Reference	1236
10.359	dcm::network::UserInformation Class Reference	1237
10.359.1	Detailed Description	1238
10.359.2	Constructor & Destructor Documentation	1238
10.359.2.1	UserInformation()	1238
10.359.2.2	~UserInformation()	1238
10.359.3	Member Function Documentation	1238
10.359.3.1	AddRoleSelectionSub()	1238
10.359.3.2	AddSOPClassExtendedNegociationSub()	1239
10.359.3.3	GetMaximumLengthSub() [1/2]	1239
10.359.3.4	GetMaximumLengthSub() [2/2]	1239
10.359.3.5	operator=()	1239
10.359.3.6	Print()	1239
10.359.3.7	Read()	1239
10.359.3.8	Size()	1240
10.359.3.9	Write()	1240
10.360	dcm::UUIDGenerator Class Reference	1240



10.360.1 Detailed Description . . . . .	1240
10.360.2 Member Function Documentation . . . . .	1240
10.360.2.1 Generate() . . . . .	1241
10.360.2.2 IsValid() . . . . .	1241
10.360.3 dcm::Validate Class Reference . . . . .	1241
10.361.1 Detailed Description . . . . .	1242
10.361.2 Constructor & Destructor Documentation . . . . .	1242
10.361.2.1 Validate() . . . . .	1242
10.361.2.2 ~Validate() . . . . .	1242
10.361.3 Member Function Documentation . . . . .	1242
10.361.3.1 GetValidatedFile() . . . . .	1242
10.361.3.2 SetFile() . . . . .	1243
10.361.3.3 Validation() . . . . .	1243
10.361.4 Member Data Documentation . . . . .	1243
10.361.4.1 F . . . . .	1243
10.361.4.2 V . . . . .	1243
10.362.0 dcm::Value Class Reference . . . . .	1244
10.362.1 Detailed Description . . . . .	1245
10.362.2 Constructor & Destructor Documentation . . . . .	1245
10.362.2.1 Value() . . . . .	1245
10.362.2.2 ~Value() . . . . .	1245
10.362.3 Member Function Documentation . . . . .	1245
10.362.3.1 Clear() . . . . .	1245
10.362.3.2 GetLength() . . . . .	1246
10.362.3.3 operator==( ) . . . . .	1246
10.362.3.4 SetLength() . . . . .	1246
10.362.3.5 SetLengthOnly() . . . . .	1246
10.362.4 Friends And Related Function Documentation . . . . .	1246

10.362.4.1DataElement . . . . .	1246
10.363dcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . . .	1247
10.363.1Detailed Description . . . . .	1247
10.363.2Member Function Documentation . . . . .	1247
10.363.2.1Read() . . . . .	1247
10.363.2.2Write() . . . . .	1247
10.364dcm::MrProtocol::Vector3 Struct Reference . . . . .	1248
10.364.1Member Data Documentation . . . . .	1248
10.364.1.1dCor . . . . .	1248
10.364.1.2dSag . . . . .	1248
10.364.1.3dTra . . . . .	1248
10.365dcm::Version Class Reference . . . . .	1248
10.365.1Detailed Description . . . . .	1249
10.365.2Constructor & Destructor Documentation . . . . .	1249
10.365.2.1Version() . . . . .	1249
10.365.2.2~Version() . . . . .	1249
10.365.3Member Function Documentation . . . . .	1249
10.365.3.1GetBuildVersion() . . . . .	1250
10.365.3.2GetMajorVersion() . . . . .	1250
10.365.3.3GetMinorVersion() . . . . .	1250
10.365.3.4GetVersion() . . . . .	1250
10.365.3.5Print() . . . . .	1250
10.365.4Friends And Related Function Documentation . . . . .	1250
10.365.4.1operator<< . . . . .	1250
10.366dcm::VL Class Reference . . . . .	1251
10.366.1Detailed Description . . . . .	1252
10.366.2Member Typedef Documentation . . . . .	1252
10.366.2.1Type . . . . .	1252

10.366.3	Constructor & Destructor Documentation	1252
10.366.3.1	VL()	1252
10.366.4	Member Function Documentation	1252
10.366.4.1	GetLength()	1252
10.366.4.2	GetVL16Max()	1253
10.366.4.3	GetVL32Max()	1253
10.366.4.4	IsOdd()	1253
10.366.4.5	IsUndefined()	1253
10.366.4.6	operator uint32_t()	1253
10.366.4.7	operator++() [1/2]	1253
10.366.4.8	operator++() [2/2]	1253
10.366.4.9	operator+=()	1254
10.366.4.10	Read()	1254
10.366.4.11	Read16()	1254
10.366.4.12	SetToUndefined()	1254
10.366.4.13	Write()	1254
10.366.4.14	Write16()	1254
10.366.5	Friends And Related Function Documentation	1255
10.366.5.1	operator<<	1255
10.367	dcm::VM Class Reference	1255
10.367.1	Detailed Description	1256
10.367.2	Member Enumeration Documentation	1257
10.367.2.1	VMType	1257
10.367.3	Constructor & Destructor Documentation	1258
10.367.3.1	VM()	1258
10.367.4	Member Function Documentation	1258
10.367.4.1	Compatible()	1258
10.367.4.2	GetIndex()	1258

10.367.4.3GetLength()	1258
10.367.4.4GetNumberOfElementsFromArray()	1258
10.367.4.5GetVMString()	1259
10.367.4.6GetVMType()	1259
10.367.4.7GetVMTypeFromLength()	1259
10.367.4.8IsValid()	1259
10.367.4.9operator VMType()	1259
10.367.5Friends And Related Function Documentation	1259
10.367.5.1operator<<	1260
10.369dcm::VMToLength< T > Struct Template Reference	1260
10.369dcm::VR Class Reference	1260
10.369.Detailed Description	1262
10.369.1Member Enumeration Documentation	1262
10.369.2.1VRType	1262
10.369.3Constructor & Destructor Documentation	1263
10.369.3.1VR()	1263
10.369.4Member Function Documentation	1263
10.369.4.1CanDisplay()	1264
10.369.4.2Compatible()	1264
10.369.4.3GetLength() [1/2]	1264
10.369.4.4GetLength() [2/2]	1264
10.369.4.5GetSize()	1264
10.369.4.6GetSizeof()	1264
10.369.4.7GetVRString()	1265
10.369.4.8GetVRStringFromFile()	1265
10.369.4.9GetVRType()	1265
10.369.4.10GetVRTypeFromFile()	1265
10.369.4.11IsASCII()	1265

10.369.4.12ASCII2()	1265
10.369.4.13Binary()	1266
10.369.4.14Binary2()	1266
10.369.4.15Dual()	1266
10.369.4.16Swap()	1266
10.369.4.17Valid() [1/2]	1266
10.369.4.18Valid() [2/2]	1266
10.369.4.19VRFile()	1267
10.369.4.20operator VRType()	1267
10.369.4.21Read()	1267
10.369.4.22Write()	1267
10.369.5Friends And Related Function Documentation	1267
10.369.5.1operator<<	1267
10.370dcm::VR16ExplicitDataElement Class Reference	1268
10.370.1Detailed Description	1269
10.370.2Member Function Documentation	1269
10.370.2.1GetLength()	1269
10.370.2.2Read()	1269
10.370.2.3ReadPreValue()	1269
10.370.2.4ReadValue()	1270
10.370.2.5ReadWithLength()	1270
10.370dcm::VRToEncoding< T > Struct Template Reference	1270
10.370dcm::VRToType< T > Struct Template Reference	1270
10.372.1Detailed Description	1271
10.370dcm::VRVLSIZE< T > Class Template Reference	1271
10.374dcm::VRVLSIZE< 0 > Class Template Reference	1271
10.374.1Member Function Documentation	1271
10.374.1.1Read()	1271

10.374.1.2Write()	1272
10.375dcm::VRVLSize< 1 > Class Template Reference	1272
10.375.1Member Function Documentation	1272
10.375.1.1Read()	1272
10.375.1.2Write()	1272
10.376tkGDCMImageReader Class Reference	1273
10.376.1Detailed Description	1275
10.376.2Constructor & Destructor Documentation	1275
10.376.2.1vtkGDCMImageReader()	1275
10.376.2.2~vtkGDCMImageReader()	1276
10.376.3Member Function Documentation	1276
10.376.3.1CanReadFile()	1276
10.376.3.2ExecuteData()	1276
10.376.3.3ExecuteInformation()	1276
10.376.3.4FillMedicalImageInformation()	1276
10.376.3.5GetDescriptiveName()	1276
10.376.3.6GetFileExtensions()	1277
10.376.3.7GetIconImage()	1277
10.376.3.8GetOverlay()	1277
10.376.3.9LoadSingleFile()	1277
10.376.3.10New()	1277
10.376.3.11PrintSelf()	1278
10.376.3.12RequestDataCompat()	1278
10.376.3.13RequestInformationCompat()	1278
10.376.3.14SetCurve()	1278
10.376.3.15SetFileNames()	1278
10.376.3.16SetFilePattern()	1278
10.376.3.17SetFilePrefix()	1279

10.376.3.18	<a href="#">SetMedicalImageProperties()</a>	1279
10.376.3.19	<a href="#">tkBooleanMacro()</a> [1/5]	1279
10.376.3.20	<a href="#">tkBooleanMacro()</a> [2/5]	1279
10.376.3.21	<a href="#">tkBooleanMacro()</a> [3/5]	1279
10.376.3.22	<a href="#">tkBooleanMacro()</a> [4/5]	1279
10.376.3.23	<a href="#">tkBooleanMacro()</a> [5/5]	1280
10.376.3.24	<a href="#">tkGetMacro()</a> [1/11]	1280
10.376.3.25	<a href="#">tkGetMacro()</a> [2/11]	1280
10.376.3.26	<a href="#">tkGetMacro()</a> [3/11]	1280
10.376.3.27	<a href="#">tkGetMacro()</a> [4/11]	1280
10.376.3.28	<a href="#">tkGetMacro()</a> [5/11]	1280
10.376.3.29	<a href="#">tkGetMacro()</a> [6/11]	1281
10.376.3.30	<a href="#">tkGetMacro()</a> [7/11]	1281
10.376.3.31	<a href="#">tkGetMacro()</a> [8/11]	1281
10.376.3.32	<a href="#">tkGetMacro()</a> [9/11]	1281
10.376.3.33	<a href="#">tkGetMacro()</a> [10/11]	1281
10.376.3.34	<a href="#">tkGetMacro()</a> [11/11]	1281
10.376.3.35	<a href="#">tkGetObjectMacro()</a> [1/4]	1282
10.376.3.36	<a href="#">tkGetObjectMacro()</a> [2/4]	1282
10.376.3.37	<a href="#">tkGetObjectMacro()</a> [3/4]	1282
10.376.3.38	<a href="#">tkGetObjectMacro()</a> [4/4]	1282
10.376.3.39	<a href="#">tkGetStringMacro()</a> [1/2]	1282
10.376.3.40	<a href="#">tkGetStringMacro()</a> [2/2]	1282
10.376.3.41	<a href="#">tkGetVector3Macro()</a>	1283
10.376.3.42	<a href="#">tkGetVector6Macro()</a>	1283
10.376.3.43	<a href="#">tkSetMacro()</a> [1/4]	1283
10.376.3.44	<a href="#">tkSetMacro()</a> [2/4]	1283
10.376.3.45	<a href="#">tkSetMacro()</a> [3/4]	1283

10.376.3.46	SetMacro() [ 4 / 4 ] . . . . .	1283
10.376.3.47	SetVector6Macro() . . . . .	1284
10.376.3.48	TypeMacro() . . . . .	1284
10.376.4	Member Data Documentation . . . . .	1284
10.376.4.1	ApplyInverseVideo . . . . .	1284
10.376.4.2	ApplyLookupTable . . . . .	1284
10.376.4.3	ApplyPlanarConfiguration . . . . .	1284
10.376.4.4	ApplyShiftScale . . . . .	1284
10.376.4.5	ApplyYBRToRGB . . . . .	1285
10.376.4.6	Curve . . . . .	1285
10.376.4.7	DirectionCosines . . . . .	1285
10.376.4.8	FileNames . . . . .	1285
10.376.4.9	ForceRescale . . . . .	1285
10.376.4.10	IconDataScalarType . . . . .	1285
10.376.4.11	IconImageDataExtent . . . . .	1285
10.376.4.12	IconNumberOfScalarComponents . . . . .	1286
10.376.4.13	ImageFormat . . . . .	1286
10.376.4.14	ImageOrientationPatient . . . . .	1286
10.376.4.15	ImagePositionPatient . . . . .	1286
10.376.4.16	LoadIconImage . . . . .	1286
10.376.4.17	LoadOverlays . . . . .	1286
10.376.4.18	LossyFlag . . . . .	1286
10.376.4.19	MedicalImageProperties . . . . .	1287
10.376.4.20	NumberOfIconImages . . . . .	1287
10.376.4.21	NumberOfOverlays . . . . .	1287
10.376.4.22	PlanarConfiguration . . . . .	1287
10.376.4.23	Scale . . . . .	1287
10.376.4.24	Shift . . . . .	1287



10.377.1 <a href="#">vtkGDCMImageReader2 Class Reference</a> . . . . .	1288
10.377.1.1 <a href="#">Detailed Description</a> . . . . .	1290
10.377.1.2 <a href="#">Constructor &amp; Destructor Documentation</a> . . . . .	1290
10.377.1.2.1 <a href="#">vtkGDCMImageReader2()</a> . . . . .	1290
10.377.1.2.2 <a href="#">~vtkGDCMImageReader2()</a> . . . . .	1290
10.377.1.3 <a href="#">Member Function Documentation</a> . . . . .	1291
10.377.1.3.1 <a href="#">CanReadFile()</a> . . . . .	1291
10.377.1.3.2 <a href="#">FillMedicalImageInformation()</a> . . . . .	1291
10.377.1.3.3 <a href="#">GetDescriptiveName()</a> . . . . .	1291
10.377.1.3.4 <a href="#">GetFileExtensions()</a> . . . . .	1291
10.377.1.3.5 <a href="#">GetIconImage()</a> . . . . .	1291
10.377.1.3.6 <a href="#">GetIconImagePort()</a> . . . . .	1291
10.377.1.3.7 <a href="#">GetOverlay()</a> . . . . .	1292
10.377.1.3.8 <a href="#">GetOverlayPort()</a> . . . . .	1292
10.377.1.3.9 <a href="#">LoadSingleFile()</a> . . . . .	1292
10.377.1.3.10 <a href="#">New()</a> . . . . .	1292
10.377.1.3.11 <a href="#">PrintSelf()</a> . . . . .	1292
10.377.1.3.12 <a href="#">ProcessRequest()</a> . . . . .	1293
10.377.1.3.13 <a href="#">RequestData()</a> . . . . .	1293
10.377.1.3.14 <a href="#">RequestDataCompat()</a> . . . . .	1293
10.377.1.3.15 <a href="#">RequestInformation()</a> . . . . .	1293
10.377.1.3.16 <a href="#">RequestInformationCompat()</a> . . . . .	1293
10.377.1.3.17 <a href="#">SetCurve()</a> . . . . .	1293
10.377.1.3.18 <a href="#">SetFilePattern()</a> . . . . .	1294
10.377.1.3.19 <a href="#">SetFilePrefix()</a> . . . . .	1294
10.377.1.3.20 <a href="#">SetMedicalImageProperties()</a> . . . . .	1294
10.377.1.3.21 <a href="#">vtkBooleanMacro()</a> [1/5] . . . . .	1294
10.377.1.3.22 <a href="#">vtkBooleanMacro()</a> [2/5] . . . . .	1294

10.377.3.23kBooleanMacro() [3/5]	1294
10.377.3.24kBooleanMacro() [4/5]	1295
10.377.3.25kBooleanMacro() [5/5]	1295
10.377.3.26kGetMacro() [1/11]	1295
10.377.3.27kGetMacro() [2/11]	1295
10.377.3.28kGetMacro() [3/11]	1295
10.377.3.29kGetMacro() [4/11]	1295
10.377.3.30kGetMacro() [5/11]	1296
10.377.3.31kGetMacro() [6/11]	1296
10.377.3.32kGetMacro() [7/11]	1296
10.377.3.33kGetMacro() [8/11]	1296
10.377.3.34kGetMacro() [9/11]	1296
10.377.3.35kGetMacro() [10/11]	1296
10.377.3.36kGetMacro() [11/11]	1297
10.377.3.37kGetObjectMacro() [1/2]	1297
10.377.3.38kGetObjectMacro() [2/2]	1297
10.377.3.39kGetStringMacro() [1/2]	1297
10.377.3.40kGetStringMacro() [2/2]	1297
10.377.3.41kGetVector3Macro()	1297
10.377.3.42kGetVector6Macro()	1298
10.377.3.43kSetMacro() [1/4]	1298
10.377.3.44kSetMacro() [2/4]	1298
10.377.3.45kSetMacro() [3/4]	1298
10.377.3.46kSetMacro() [4/4]	1298
10.377.3.47kSetVector6Macro()	1298
10.377.3.48kTypeMacro()	1299
10.377.4Member Data Documentation	1299
10.377.4.1ApplyInverseVideo	1299

10.377.4.2	<a href="#">ApplyLookupTable</a>	1299
10.377.4.3	<a href="#">ApplyPlanarConfiguration</a>	1299
10.377.4.4	<a href="#">ApplyShiftScale</a>	1299
10.377.4.5	<a href="#">ApplyYBRToRGB</a>	1299
10.377.4.6	<a href="#">Curve</a>	1300
10.377.4.7	<a href="#">DirectionCosines</a>	1300
10.377.4.8	<a href="#">ForceRescale</a>	1300
10.377.4.9	<a href="#">IconDataScalarType</a>	1300
10.377.4.10	<a href="#">IconImageDataExtent</a>	1300
10.377.4.11	<a href="#">IconNumberOfScalarComponents</a>	1300
10.377.4.12	<a href="#">ImageFormat</a>	1300
10.377.4.13	<a href="#">ImageOrientationPatient</a>	1301
10.377.4.14	<a href="#">ImagePositionPatient</a>	1301
10.377.4.15	<a href="#">LoadIconImage</a>	1301
10.377.4.16	<a href="#">LoadOverlays</a>	1301
10.377.4.17	<a href="#">LossyFlag</a>	1301
10.377.4.18	<a href="#">NumberOfIconImages</a>	1301
10.377.4.19	<a href="#">NumberOfOverlays</a>	1301
10.377.4.20	<a href="#">PlanarConfiguration</a>	1302
10.377.4.21	<a href="#">Scale</a>	1302
10.377.4.22	<a href="#">Shift</a>	1302
10.378	<a href="#">vtkGDCMImageWriter Class Reference</a>	1302
10.378.1	<a href="#">Detailed Description</a>	1304
10.378.2	<a href="#">Member Enumeration Documentation</a>	1304
10.378.2.1	<a href="#">CompressionTypes</a>	1304
10.378.3	<a href="#">Constructor &amp; Destructor Documentation</a>	1305
10.378.3.1	<a href="#">vtkGDCMImageWriter()</a>	1305
10.378.3.2	<a href="#">~vtkGDCMImageWriter()</a>	1305

10.378.4	Member Function Documentation	1305
10.378.4.1	GetDescriptiveName()	1305
10.378.4.2	GetFileExtensions()	1305
10.378.4.3	GetFileName()	1305
10.378.4.4	New()	1306
10.378.4.5	PrintSelf()	1306
10.378.4.6	SetDirectionCosines()	1306
10.378.4.7	SetDirectionCosinesFromImageOrientationPatient()	1306
10.378.4.8	SetFileNames()	1306
10.378.4.9	SetMedicalImageProperties()	1307
10.378.4.10	BooleanMacro() [1/2]	1307
10.378.4.11	BooleanMacro() [2/2]	1307
10.378.4.12	GetMacro() [1/7]	1307
10.378.4.13	GetMacro() [2/7]	1307
10.378.4.14	GetMacro() [3/7]	1308
10.378.4.15	GetMacro() [4/7]	1308
10.378.4.16	GetMacro() [5/7]	1308
10.378.4.17	GetMacro() [6/7]	1308
10.378.4.18	GetMacro() [7/7]	1308
10.378.4.19	GetObjectMacro() [1/3]	1308
10.378.4.20	GetObjectMacro() [2/3]	1309
10.378.4.21	GetObjectMacro() [3/3]	1309
10.378.4.22	GetStringMacro() [1/2]	1309
10.378.4.23	GetStringMacro() [2/2]	1309
10.378.4.24	SetMacro() [1/7]	1309
10.378.4.25	SetMacro() [2/7]	1309
10.378.4.26	SetMacro() [3/7]	1310
10.378.4.27	SetMacro() [4/7]	1310

10.378.4.28	<a href="#">vtkSetMacro() [5/7]</a>	1310
10.378.4.29	<a href="#">vtkSetMacro() [6/7]</a>	1310
10.378.4.30	<a href="#">vtkSetMacro() [7/7]</a>	1310
10.378.4.31	<a href="#">vtkSetStringMacro() [1/2]</a>	1310
10.378.4.32	<a href="#">vtkSetStringMacro() [2/2]</a>	1311
10.378.4.33	<a href="#">vtkTypeMacro()</a>	1311
10.378.4.34	<a href="#">Write()</a>	1311
10.378.4.35	<a href="#">WriteGDCMData()</a>	1311
10.378.4.36	<a href="#">WriteSlice()</a>	1311
10.379	<a href="#">vtkGDCMMedicalImageProperties Class Reference</a>	1312
10.379.1	<a href="#">Constructor &amp; Destructor Documentation</a>	1313
10.379.1.1	<a href="#">vtkGDCMMedicalImageProperties()</a>	1313
10.379.1.2	<a href="#">~vtkGDCMMedicalImageProperties()</a>	1313
10.379.2	<a href="#">Member Function Documentation</a>	1313
10.379.2.1	<a href="#">Clear()</a>	1313
10.379.2.2	<a href="#">GetFile()</a>	1313
10.379.2.3	<a href="#">New()</a>	1314
10.379.2.4	<a href="#">PrintSelf()</a>	1314
10.379.2.5	<a href="#">PushBackFile()</a>	1314
10.379.2.6	<a href="#">vtkTypeMacro()</a>	1314
10.379.3	<a href="#">Friends And Related Function Documentation</a>	1314
10.379.3.1	<a href="#">vtkGDCMImageReader</a>	1314
10.379.3.2	<a href="#">vtkGDCMImageReader2</a>	1314
10.379.3.3	<a href="#">vtkGDCMImageWriter</a>	1315
10.380	<a href="#">vtkGDCMPolyDataReader Class Reference</a>	1315
10.380.1	<a href="#">Detailed Description</a>	1316
10.380.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1316
10.380.2.1	<a href="#">vtkGDCMPolyDataReader()</a>	1317

10.380.2.2~vtkGDCMPolyDataReader()	1317
10.380.3Member Function Documentation	1317
10.380.3.1FillMedicalImageInformation()	1317
10.380.3.2New()	1317
10.380.3.3PrintSelf()	1317
10.380.3.4RequestData()	1318
10.380.3.5RequestData_HemodynamicWaveformStorage()	1318
10.380.3.6RequestData_RTStructureSetStorage()	1318
10.380.3.7RequestInformation()	1318
10.380.3.8RequestInformation_HemodynamicWaveformStorage()	1318
10.380.3.9RequestInformation_RTStructureSetStorage()	1318
10.380.3.10GetObjectMacro() [1/2]	1319
10.380.3.11GetObjectMacro() [2/2]	1319
10.380.3.12GetStringMacro()	1319
10.380.3.13SetStringMacro()	1319
10.380.3.14TypeMacro()	1319
10.380.4Member Data Documentation	1319
10.380.4.1FileName	1319
10.380.4.2MedicalImageProperties	1320
10.380.4.3RTStructSetProperties	1320
10.381.1vtkGDCMPolyDataWriter Class Reference	1320
10.381.1.1Detailed Description	1322
10.381.2Constructor & Destructor Documentation	1322
10.381.2.1vtkGDCMPolyDataWriter()	1322
10.381.2.2~vtkGDCMPolyDataWriter()	1322
10.381.3Member Function Documentation	1322
10.381.3.1InitializeRTStructSet()	1322
10.381.3.2New()	1323

10.381.3.3PrintSelf()	1323
10.381.3.4SetMedicalImageProperties()	1323
10.381.3.5SetNumberOfInputPorts()	1323
10.381.3.6SetRTStructSetProperties()	1324
10.381.3.7vtkTypeMacro()	1324
10.381.3.8WriteData()	1324
10.381.3.9WriteRTSTRUCTData()	1324
10.381.3.10WriteRTSTRUCTInfo()	1324
10.381.4Member Data Documentation	1324
10.381.4.1MedicalImageProperties	1325
10.381.4.2RTStructSetProperties	1325
10.382.1GDCMTesting Class Reference	1325
10.382.1Detailed Description	1326
10.382.2Member Typedef Documentation	1326
10.382.2.1MD5MetalImagesType	1326
10.382.3Constructor & Destructor Documentation	1326
10.382.3.1vtkGDCMTesting()	1327
10.382.3.2~vtkGDCMTesting()	1327
10.382.4Member Function Documentation	1327
10.382.4.1GetGDCMDataRoot()	1327
10.382.4.2GetMD5MetalImage()	1327
10.382.4.3GetMHDMD5FromFile()	1327
10.382.4.4GetNumberOfMD5MetalImages()	1328
10.382.4.5GetRAWMD5FromFile()	1328
10.382.4.6GetVTKDataRoot()	1328
10.382.4.7New()	1328
10.382.4.8PrintSelf()	1328
10.382.4.9vtkTypeMacro()	1329

10.383. <del>1</del> tkGDCMThreadedImageReader Class Reference . . . . .	1329
10.383.1. <del>1</del> Constructor & Destructor Documentation . . . . .	1330
10.383.1.1. <del>1</del> tkGDCMThreadedImageReader() . . . . .	1331
10.383.1.2. <del>1</del> ~tkGDCMThreadedImageReader() . . . . .	1331
10.383.2. <del>1</del> Member Function Documentation . . . . .	1331
10.383.2.1. <del>1</del> ExecuteData() . . . . .	1331
10.383.2.2. <del>1</del> ExecuteInformation() . . . . .	1331
10.383.2.3. <del>1</del> New() . . . . .	1331
10.383.2.4. <del>1</del> PrintSelf() . . . . .	1331
10.383.2.5. <del>1</del> ReadFiles() . . . . .	1332
10.383.2.6. <del>1</del> RequestDataCompat() . . . . .	1332
10.383.2.7. <del>1</del> tkBooleanMacro() . . . . .	1332
10.383.2.8. <del>1</del> tkGetMacro() . . . . .	1332
10.383.2.9. <del>1</del> tkSetMacro() [1/3] . . . . .	1332
10.383.2.10. <del>1</del> tkSetMacro() [2/3] . . . . .	1332
10.383.2.11. <del>1</del> tkSetMacro() [3/3] . . . . .	1333
10.383.2.12. <del>1</del> tkTypeMacro() . . . . .	1333
10.384. <del>1</del> tkGDCMThreadedImageReader2 Class Reference . . . . .	1333
10.384.1. <del>1</del> Constructor & Destructor Documentation . . . . .	1335
10.384.1.1. <del>1</del> tkGDCMThreadedImageReader2() . . . . .	1335
10.384.1.2. <del>1</del> ~tkGDCMThreadedImageReader2() . . . . .	1335
10.384.2. <del>1</del> Member Function Documentation . . . . .	1335
10.384.2.1. <del>1</del> GetFileName() . . . . .	1335
10.384.2.2. <del>1</del> New() . . . . .	1335
10.384.2.3. <del>1</del> PrintSelf() . . . . .	1336
10.384.2.4. <del>1</del> RequestInformation() . . . . .	1336
10.384.2.5. <del>1</del> SetFileName() . . . . .	1336
10.384.2.6. <del>1</del> SetFileNames() . . . . .	1336



10.384.2.7SplitExtent()	1336
10.384.2.8ThreadedRequestData()	1337
10.384.2.9vtkBooleanMacro() [1/3]	1337
10.384.2.10vtkBooleanMacro() [2/3]	1337
10.384.2.11vtkBooleanMacro() [3/3]	1337
10.384.2.12vtkGetMacro() [1/8]	1337
10.384.2.13vtkGetMacro() [2/8]	1338
10.384.2.14vtkGetMacro() [3/8]	1338
10.384.2.15vtkGetMacro() [4/8]	1338
10.384.2.16vtkGetMacro() [5/8]	1338
10.384.2.17vtkGetMacro() [6/8]	1338
10.384.2.18vtkGetMacro() [7/8]	1338
10.384.2.19vtkGetMacro() [8/8]	1339
10.384.2.20vtkGetObjectMacro()	1339
10.384.2.21vtkGetVector3Macro() [1/2]	1339
10.384.2.22vtkGetVector3Macro() [2/2]	1339
10.384.2.23vtkGetVector6Macro()	1339
10.384.2.24vtkSetMacro() [1/7]	1339
10.384.2.25vtkSetMacro() [2/7]	1340
10.384.2.26vtkSetMacro() [3/7]	1340
10.384.2.27vtkSetMacro() [4/7]	1340
10.384.2.28vtkSetMacro() [5/7]	1340
10.384.2.29vtkSetMacro() [6/7]	1340
10.384.2.30vtkSetMacro() [7/7]	1340
10.384.2.31vtkSetVector3Macro() [1/2]	1341
10.384.2.32vtkSetVector3Macro() [2/2]	1341
10.384.2.33vtkSetVector6Macro()	1341
10.384.2.34vtkTypeMacro()	1341

10.385.1	vtkImageColorViewer Class Reference . . . . .	1342
10.385.1	Detailed Description . . . . .	1344
10.385.2	Member Enumeration Documentation . . . . .	1344
10.385.2.1	anonymous enum . . . . .	1344
10.385.3	Constructor & Destructor Documentation . . . . .	1345
10.385.3.1	vtkImageColorViewer() . . . . .	1345
10.385.3.2	~vtkImageColorViewer() . . . . .	1345
10.385.4	Member Function Documentation . . . . .	1345
10.385.4.1	AddInput() . . . . .	1345
10.385.4.2	AddInputConnection() . . . . .	1345
10.385.4.3	GetColorLevel() . . . . .	1346
10.385.4.4	GetColorWindow() . . . . .	1346
10.385.4.5	GetInput() . . . . .	1346
10.385.4.6	GetOffScreenRendering() . . . . .	1346
10.385.4.7	GetOverlayVisibility() . . . . .	1346
10.385.4.8	GetPosition() . . . . .	1346
10.385.4.9	GetSize() . . . . .	1346
10.385.4.10	GetSliceMax() . . . . .	1347
10.385.4.10	GetSliceMin() . . . . .	1347
10.385.4.10	GetSliceRange() [1/3] . . . . .	1347
10.385.4.10	GetSliceRange() [2/3] . . . . .	1347
10.385.4.10	GetSliceRange() [3/3] . . . . .	1347
10.385.4.10	GetWindowName() . . . . .	1347
10.385.4.10	InstallPipeline() . . . . .	1347
10.385.4.10	New() . . . . .	1348
10.385.4.10	PrintSelf() . . . . .	1348
10.385.4.10	Render() . . . . .	1348
10.385.4.20	SetColorLevel() . . . . .	1348

10.385.4.28	<a href="#">SetColorWindow()</a>	1348
10.385.4.29	<a href="#">SetDisplayId()</a>	1349
10.385.4.29	<a href="#">SetInput()</a>	1349
10.385.4.29	<a href="#">SetInputConnection()</a>	1349
10.385.4.29	<a href="#">SetOffScreenRendering()</a>	1349
10.385.4.29	<a href="#">SetOverlayVisibility()</a>	1349
10.385.4.29	<a href="#">SetParentId()</a>	1349
10.385.4.29	<a href="#">SetPosition() [1/2]</a>	1350
10.385.4.29	<a href="#">SetPosition() [2/2]</a>	1350
10.385.4.30	<a href="#">SetRenderer()</a>	1350
10.385.4.30	<a href="#">SetRenderWindow()</a>	1350
10.385.4.30	<a href="#">SetSize() [1/2]</a>	1350
10.385.4.30	<a href="#">SetSize() [2/2]</a>	1351
10.385.4.30	<a href="#">SetSlice()</a>	1351
10.385.4.35	<a href="#">SetSliceOrientation()</a>	1351
10.385.4.36	<a href="#">SetSliceOrientationToXY()</a>	1351
10.385.4.36	<a href="#">SetSliceOrientationToXZ()</a>	1351
10.385.4.36	<a href="#">SetSliceOrientationToYZ()</a>	1351
10.385.4.39	<a href="#">SetupInteractor()</a>	1352
10.385.4.40	<a href="#">SetWindowId()</a>	1352
10.385.4.41	<a href="#">InstallPipeline()</a>	1352
10.385.4.42	<a href="#">UpdateDisplayExtent()</a>	1352
10.385.4.43	<a href="#">UpdateOrientation()</a>	1352
10.385.4.44	<a href="#">WTK_LEGACY() [1/4]</a>	1352
10.385.4.45	<a href="#">WTK_LEGACY() [2/4]</a>	1353
10.385.4.46	<a href="#">WTK_LEGACY() [3/4]</a>	1353
10.385.4.47	<a href="#">WTK_LEGACY() [4/4]</a>	1353
10.385.4.48	<a href="#">BooleanMacro()</a>	1353

10.385.4.40	<a href="#">vtkGetMacro() [1/2]</a>	1353
10.385.4.50	<a href="#">vtkGetMacro() [2/2]</a>	1353
10.385.4.51	<a href="#">vtkGetObjectMacro() [1/5]</a>	1354
10.385.4.52	<a href="#">vtkGetObjectMacro() [2/5]</a>	1354
10.385.4.53	<a href="#">vtkGetObjectMacro() [3/5]</a>	1354
10.385.4.54	<a href="#">vtkGetObjectMacro() [4/5]</a>	1354
10.385.4.55	<a href="#">vtkGetObjectMacro() [5/5]</a>	1354
10.385.4.56	<a href="#">vtkTypeMacro()</a>	1354
10.385.5	<a href="#">Friends And Related Function Documentation</a>	1355
10.385.5.1	<a href="#">vtkImageColorViewerCallback</a>	1355
10.385.6	<a href="#">Member Data Documentation</a>	1355
10.385.6.1	<a href="#">FirstRender</a>	1355
10.385.6.2	<a href="#">ImageActor</a>	1355
10.385.6.3	<a href="#">Interactor</a>	1355
10.385.6.4	<a href="#">InteractorStyle</a>	1355
10.385.6.5	<a href="#">OverlayImageActor</a>	1355
10.385.6.6	<a href="#">Renderer</a>	1356
10.385.6.7	<a href="#">RenderWindow</a>	1356
10.385.6.8	<a href="#">Slice</a>	1356
10.385.6.9	<a href="#">SliceOrientation</a>	1356
10.385.6.10	<a href="#">WindowLevel</a>	1356
10.386	<a href="#">vtkImageMapToColors16 Class Reference</a>	1357
10.386.1	<a href="#">Constructor &amp; Destructor Documentation</a>	1358
10.386.1.1	<a href="#">vtkImageMapToColors16()</a>	1358
10.386.1.2	<a href="#">~vtkImageMapToColors16()</a>	1358
10.386.2	<a href="#">Member Function Documentation</a>	1359
10.386.2.1	<a href="#">GetMTime()</a>	1359
10.386.2.2	<a href="#">New()</a>	1359

10.386.2.3PrintSelf()	1359
10.386.2.4RequestData()	1359
10.386.2.5RequestInformation()	1359
10.386.2.6SetLookupTable()	1360
10.386.2.7SetOutputFormatToLuminance()	1360
10.386.2.8SetOutputFormatToLuminanceAlpha()	1360
10.386.2.9SetOutputFormatToRGB()	1360
10.386.2.10SetOutputFormatToRGBA()	1360
10.386.2.11ThreadedRequestData()	1360
10.386.2.12tkBooleanMacro()	1361
10.386.2.13tkGetMacro() [1/3]	1361
10.386.2.14tkGetMacro() [2/3]	1361
10.386.2.15tkGetMacro() [3/3]	1361
10.386.2.16tkGetObjectMacro()	1361
10.386.2.17tkSetMacro() [1/3]	1361
10.386.2.18tkSetMacro() [2/3]	1362
10.386.2.19tkSetMacro() [3/3]	1362
10.386.2.20tkTypeMacro()	1362
10.386.3Member Data Documentation	1362
10.386.3.1ActiveComponent	1362
10.386.3.2DataWasPassed	1362
10.386.3.3LookupTable	1362
10.386.3.4OutputFormat	1363
10.386.3.5PassAlphaToOutput	1363
10.387.vtkImageMapToWindowLevelColors2 Class Reference	1363
10.387.1Constructor & Destructor Documentation	1364
10.387.1.1vtkImageMapToWindowLevelColors2()	1364
10.387.1.2~vtkImageMapToWindowLevelColors2()	1364

10.387.2	Member Function Documentation	1365
10.387.2.1	New()	1365
10.387.2.2	PrintSelf()	1365
10.387.2.3	RequestData()	1365
10.387.2.4	RequestInformation()	1365
10.387.2.5	ThreadedRequestData()	1365
10.387.2.6	vtkGetMacro() [1/2]	1366
10.387.2.7	vtkGetMacro() [2/2]	1366
10.387.2.8	vtkSetMacro() [1/2]	1366
10.387.2.9	vtkSetMacro() [2/2]	1366
10.387.2.10	vtkTypeMacro()	1366
10.387.3	Member Data Documentation	1366
10.387.3.1	Level	1367
10.387.3.2	Window	1367
10.388	vtkImagePlanarComponentsToComponents Class Reference	1367
10.388.1	Constructor & Destructor Documentation	1368
10.388.1.1	vtkImagePlanarComponentsToComponents()	1368
10.388.1.2	~vtkImagePlanarComponentsToComponents()	1368
10.388.2	Member Function Documentation	1368
10.388.2.1	New()	1368
10.388.2.2	PrintSelf()	1369
10.388.2.3	RequestData()	1369
10.388.2.4	vtkTypeMacro()	1369
10.389	vtkImageRGBToYBR Class Reference	1369
10.389.1	Constructor & Destructor Documentation	1370
10.389.1.1	vtkImageRGBToYBR()	1370
10.389.1.2	~vtkImageRGBToYBR()	1370
10.389.2	Member Function Documentation	1371

10.389.2.1New()	1371
10.389.2.2PrintSelf()	1371
10.389.2.3ThreadedExecute()	1371
10.389.2.4vtkTypeMacro()	1371
10.390.vtkImageYBRToRGB Class Reference	1372
10.390.1 Constructor & Destructor Documentation	1373
10.390.1.1vtkImageYBRToRGB()	1373
10.390.1.2~vtkImageYBRToRGB()	1373
10.390.2 Member Function Documentation	1373
10.390.2.1New()	1373
10.390.2.2PrintSelf()	1373
10.390.2.3ThreadedExecute()	1374
10.390.2.4vtkTypeMacro()	1374
10.391.vtkLookupTable16 Class Reference	1374
10.391.1 Constructor & Destructor Documentation	1375
10.391.1.1vtkLookupTable16()	1376
10.391.1.2~vtkLookupTable16()	1376
10.391.2 Member Function Documentation	1376
10.391.2.1Build()	1376
10.391.2.2GetPointer()	1376
10.391.2.3MapScalarsThroughTable2()	1376
10.391.2.4New()	1377
10.391.2.5PrintSelf()	1377
10.391.2.6SetNumberOfTableValues()	1377
10.391.2.7vtkTypeMacro()	1377
10.391.2.8WritePointer()	1377
10.391.3 Member Data Documentation	1377
10.391.3.1Table16	1377

10.392.1	vtkRTStructSetProperties Class Reference	1378
10.392.1	Detailed Description	1380
10.392.2	Constructor & Destructor Documentation	1380
10.392.2.1	vtkRTStructSetProperties()	1380
10.392.2.2	~vtkRTStructSetProperties()	1380
10.392.3	Member Function Documentation	1380
10.392.3.1	AddContourReferencedFrameOfReference()	1380
10.392.3.2	AddReferencedFrameOfReference()	1381
10.392.3.3	AddStructureSetROI()	1381
10.392.3.4	AddStructureSetROIObservation()	1381
10.392.3.5	Clear()	1381
10.392.3.6	DeepCopy()	1381
10.392.3.7	GetContourReferencedFrameOfReferenceClassUID()	1382
10.392.3.8	GetContourReferencedFrameOfReferenceInstanceUID()	1382
10.392.3.9	GetNumberOfContourReferencedFrameOfReferences() [1/2]	1382
10.392.3.10	GetNumberOfContourReferencedFrameOfReferences() [2/2]	1382
10.392.3.11	GetNumberOfReferencedFrameOfReferences()	1382
10.392.3.12	GetNumberOfStructureSetROIs()	1382
10.392.3.13	GetReferencedFrameOfReferenceClassUID()	1383
10.392.3.14	GetReferencedFrameOfReferenceInstanceUID()	1383
10.392.3.15	GetStructureSetObservationNumber()	1383
10.392.3.16	GetStructureSetROIDescription()	1383
10.392.3.17	GetStructureSetROIGenerationAlgorithm()	1383
10.392.3.18	GetStructureSetROIName()	1383
10.392.3.19	GetStructureSetROINumber()	1384
10.392.3.20	GetStructureSetROIObservationLabel()	1384
10.392.3.21	GetStructureSetROIRefFrameRefUID()	1384
10.392.3.22	GetStructureSetRTROIInterpretedType()	1384



10.392.3.29	New()	1384
10.392.3.29	PrintSelf()	1384
10.392.3.29	GetStringMacro() [1/9]	1385
10.392.3.29	GetStringMacro() [2/9]	1385
10.392.3.29	GetStringMacro() [3/9]	1385
10.392.3.29	GetStringMacro() [4/9]	1385
10.392.3.29	GetStringMacro() [5/9]	1385
10.392.3.29	GetStringMacro() [6/9]	1385
10.392.3.29	GetStringMacro() [7/9]	1386
10.392.3.29	GetStringMacro() [8/9]	1386
10.392.3.29	GetStringMacro() [9/9]	1386
10.392.3.30	SetStringMacro() [1/9]	1386
10.392.3.30	SetStringMacro() [2/9]	1386
10.392.3.30	SetStringMacro() [3/9]	1386
10.392.3.30	SetStringMacro() [4/9]	1387
10.392.3.30	SetStringMacro() [5/9]	1387
10.392.3.30	SetStringMacro() [6/9]	1387
10.392.3.30	SetStringMacro() [7/9]	1387
10.392.3.30	SetStringMacro() [8/9]	1387
10.392.3.30	SetStringMacro() [9/9]	1387
10.392.3.40	TypeMacro()	1388
10.392.4	Member Data Documentation	1388
10.392.4.1	Internals	1388
10.392.4.2	ReferenceFrameOfReferenceUID	1388
10.392.4.3	ReferenceSeriesInstanceUID	1388
10.392.4.4	SeriesInstanceUID	1388
10.392.4.5	SOPInstanceUID	1388
10.392.4.6	StructureSetDate	1389

10.392.4.7	StructureSetLabel	1389
10.392.4.8	StructureSetName	1389
10.392.4.9	StructureSetTime	1389
10.392.4.10	StudyInstanceUID	1389
10.393	dcm::Waveform Class Reference	1389
10.393.1	Detailed Description	1390
10.393.2	Constructor & Destructor Documentation	1390
10.393.2.1	Waveform()	1390
10.394	dcm::WLMFindQuery Class Reference	1390
10.394.1	Detailed Description	1391
10.394.2	Constructor & Destructor Documentation	1391
10.394.2.1	WLMFindQuery()	1392
10.394.3	Member Function Documentation	1392
10.394.3.1	GetAbstractSyntaxUID()	1392
10.394.3.2	GetTagListByLevel()	1392
10.394.3.3	GetValidDataSet()	1392
10.394.3.4	InitializeDataSet()	1392
10.394.3.5	ValidateQuery()	1393
10.394.4	Friends And Related Function Documentation	1393
10.394.4.1	QueryFactory	1393
10.395	dcm::Writer Class Reference	1393
10.395.1	Detailed Description	1395
10.395.2	Constructor & Destructor Documentation	1396
10.395.2.1	Writer()	1396
10.395.2.2	~Writer()	1396
10.395.3	Member Function Documentation	1396
10.395.3.1	CheckFileMetaInformationOff()	1396
10.395.3.2	CheckFileMetaInformationOn()	1397

10.395.3.3GetCheckFileMetaInformation()	1397
10.395.3.4GetFile()	1397
10.395.3.5GetStreamPtr()	1397
10.395.3.6SetCheckFileMetaInformation()	1397
10.395.3.7SetFile()	1398
10.395.3.8SetFileName()	1398
10.395.3.9SetStream()	1398
10.395.3.10SetWriteDataSetOnly()	1398
10.395.3.11Write()	1399
10.395.4Friends And Related Function Documentation	1399
10.395.4.1StreamImageWriter	1399
10.395.5Member Data Documentation	1399
10.395.5.1Ofstream	1399
10.395.5.2Stream	1399
10.396dcm::XMLDictReader Class Reference	1400
10.396.1Detailed Description	1401
10.396.2Constructor & Destructor Documentation	1401
10.396.2.1XMLDictReader()	1401
10.396.2.2~XMLDictReader()	1401
10.396.3Member Function Documentation	1401
10.396.3.1CharacterDataHandler()	1401
10.396.3.2EndElement()	1402
10.396.3.3GetDict()	1402
10.396.3.4HandleDescription()	1402
10.396.3.5HandleEntry()	1402
10.396.3.6StartElement()	1402
10.397dcm::XMLPrinter Class Reference	1403
10.397.1Member Enumeration Documentation	1404

10.397.1.1PrintStyles . . . . .	1404
10.397.2Constructor & Destructor Documentation . . . . .	1404
10.397.2.1XMLPrinter() . . . . .	1404
10.397.2.2~XMLPrinter() . . . . .	1404
10.397.3Member Function Documentation . . . . .	1404
10.397.3.1GetPrintStyle() . . . . .	1404
10.397.3.2HandleBulkData() . . . . .	1405
10.397.3.3Print() . . . . .	1405
10.397.3.4PrintDataElement() . . . . .	1405
10.397.3.5PrintDataSet() . . . . .	1405
10.397.3.6PrintSQ() . . . . .	1405
10.397.3.7SetFile() . . . . .	1406
10.397.3.8SetStyle() . . . . .	1406
10.397.4Member Data Documentation . . . . .	1406
10.397.4.1F . . . . .	1406
10.397.4.2PrintStyle . . . . .	1406
10.398dcm::XMLPrivateDictReader Class Reference . . . . .	1407
10.398.1Detailed Description . . . . .	1408
10.398.2Constructor & Destructor Documentation . . . . .	1408
10.398.2.1XMLPrivateDictReader() . . . . .	1408
10.398.2.2~XMLPrivateDictReader() . . . . .	1408
10.398.3Member Function Documentation . . . . .	1408
10.398.3.1CharacterDataHandler() . . . . .	1408
10.398.3.2EndElement() . . . . .	1409
10.398.3.3GetPrivateDict() . . . . .	1409
10.398.3.4HandleDescription() . . . . .	1409
10.398.3.5HandleEntry() . . . . .	1409
10.398.3.6StartElement() . . . . .	1409

<b>11 File Documentation</b>	<b>1411</b>
11.1 gdcmAAbortPDU.h File Reference . . . . .	1411
11.2 gdcmAAssociateACPDU.h File Reference . . . . .	1412
11.3 gdcmAAssociateRJPDU.h File Reference . . . . .	1412
11.4 gdcmAAssociateRQPDU.h File Reference . . . . .	1413
11.5 gdcmAbstractSyntax.h File Reference . . . . .	1414
11.6 gdcmAnonymizeEvent.h File Reference . . . . .	1415
11.7 gdcmAnonymizer.h File Reference . . . . .	1416
11.8 gdcmApplicationContext.h File Reference . . . . .	1417
11.9 gdcmApplicationEntity.h File Reference . . . . .	1418
11.10gdcmAReleaseRPPDU.h File Reference . . . . .	1418
11.11gdcmAReleaseRQPDU.h File Reference . . . . .	1419
11.12gdcmARTIMTimer.h File Reference . . . . .	1420
11.13gdcmASN1.h File Reference . . . . .	1421
11.14gdcmAsynchronousOperationsWindowSub.h File Reference . . . . .	1422
11.15gdcmAttribute.h File Reference . . . . .	1422
11.16gdcmAudioCodec.h File Reference . . . . .	1424
11.17gdcmBase64.h File Reference . . . . .	1424
11.18gdcmBaseCompositeMessage.h File Reference . . . . .	1425
11.19gdcmBaseNormalizedMessage.h File Reference . . . . .	1426
11.20gdcmBasePDU.h File Reference . . . . .	1427
11.21gdcmBaseQuery.h File Reference . . . . .	1428
11.22gdcmBaseRootQuery.h File Reference . . . . .	1429
11.23gdcmBasicOffsetTable.h File Reference . . . . .	1430
11.24gdcmBitmap.h File Reference . . . . .	1431
11.25gdcmBitmapToBitmapFilter.h File Reference . . . . .	1432
11.26gdcmBoxRegion.h File Reference . . . . .	1433
11.27gdcmByteBuffer.h File Reference . . . . .	1433

11.28gdcmbSwap.h File Reference . . . . .	1435
11.29gdcmbSwapFilter.h File Reference . . . . .	1435
11.30gdcmbSwapValue.h File Reference . . . . .	1436
11.31gdcmbCryptoFactory.h File Reference . . . . .	1437
11.32gdcmbCryptographicMessageSyntax.h File Reference . . . . .	1438
11.33gdcmbCEchoMessages.h File Reference . . . . .	1438
11.34gdcmbCFindMessages.h File Reference . . . . .	1439
11.35gdcmbCMoveMessages.h File Reference . . . . .	1440
11.36gdcmbCodec.h File Reference . . . . .	1441
11.37gdcmbCoder.h File Reference . . . . .	1442
11.38gdcmbCodeString.h File Reference . . . . .	1444
11.39gdcmbCommand.h File Reference . . . . .	1445
11.40gdcmbCommandDataSet.h File Reference . . . . .	1446
11.41gdcmbCompositeMessageFactory.h File Reference . . . . .	1447
11.42gdcmbCompositeNetworkFunctions.h File Reference . . . . .	1447
11.43gdcmbConstCharWrapper.h File Reference . . . . .	1448
11.44gdcmbCP246ExplicitDataElement.h File Reference . . . . .	1448
11.45gdcmbCryptoFactory.h File Reference . . . . .	1449
11.46gdcmbCryptographicMessageSyntax.h File Reference . . . . .	1450
11.47gdcmbCSAElement.h File Reference . . . . .	1451
11.48gdcmbCSAHeader.h File Reference . . . . .	1452
11.49gdcmbCSAHeaderDict.h File Reference . . . . .	1453
11.50gdcmbCSAHeaderDictEntry.h File Reference . . . . .	1454
11.51gdcmbCStoreMessages.h File Reference . . . . .	1455
11.52gdcmbCurve.h File Reference . . . . .	1456
11.53gdcmbDataElement.h File Reference . . . . .	1457
11.54gdcmbDataEvent.h File Reference . . . . .	1459
11.55gdcmbDataSet.h File Reference . . . . .	1460

11.56gdcmDataSetEvent.h File Reference . . . . .	1461
11.57gdcmDataSetHelper.h File Reference . . . . .	1461
11.58gdcmDecoder.h File Reference . . . . .	1462
11.59gdcmDefinedTerms.h File Reference . . . . .	1464
11.60gdcmDeflateStream.h File Reference . . . . .	1464
11.61gdcmDefs.h File Reference . . . . .	1465
11.62gdcmDeltaEncodingCodec.h File Reference . . . . .	1466
11.63gdcmDICOMDIR.h File Reference . . . . .	1466
11.64gdcmDICOMDIRGenerator.h File Reference . . . . .	1467
11.65gdcmDict.h File Reference . . . . .	1468
11.66gdcmDictConverter.h File Reference . . . . .	1469
11.67gdcmDictEntry.h File Reference . . . . .	1470
11.68gdcmDictPrinter.h File Reference . . . . .	1471
11.69gdcmDicts.h File Reference . . . . .	1472
11.70gdcmDIMSE.h File Reference . . . . .	1473
11.71gdcmDirectionCosines.h File Reference . . . . .	1473
11.72gdcmDirectory.h File Reference . . . . .	1474
11.73gdcmDirectoryHelper.h File Reference . . . . .	1475
11.74gdcmDummyValueGenerator.h File Reference . . . . .	1476
11.75gdcmDumper.h File Reference . . . . .	1476
11.76gdcmElement.h File Reference . . . . .	1477
11.76.1 Macro Definition Documentation . . . . .	1479
11.76.1.1 VRDS16ILLEGAL . . . . .	1479
11.77gdcmEmptyMaskGenerator.h File Reference . . . . .	1479
11.78gdcmEncapsulatedDocument.h File Reference . . . . .	1480
11.79gdcmEnumeratedValues.h File Reference . . . . .	1481
11.80gdcmEvent.h File Reference . . . . .	1481
11.80.1 Macro Definition Documentation . . . . .	1482

11.80.1.1 gdcmacroEventMacro . . . . .	1483
11.81gdcmacroException.h File Reference . . . . .	1483
11.82gdcmacroExplicitDataElement.h File Reference . . . . .	1484
11.83gdcmacroExplicitImplicitDataElement.h File Reference . . . . .	1485
11.84gdcmacroFiducials.h File Reference . . . . .	1485
11.85gdcmacroFile.h File Reference . . . . .	1486
11.86gdcmacroFileAnonymizer.h File Reference . . . . .	1487
11.87gdcmacroFileChangeTransferSyntax.h File Reference . . . . .	1488
11.88gdcmacroFileDecompressLookupTable.h File Reference . . . . .	1488
11.89gdcmacroFileDerivation.h File Reference . . . . .	1489
11.90gdcmacroFileExplicitFilter.h File Reference . . . . .	1490
11.91gdcmacroFileMetaInformation.h File Reference . . . . .	1491
11.92gdcmacroFilename.h File Reference . . . . .	1492
11.93gdcmacroFileNameEvent.h File Reference . . . . .	1492
11.94gdcmacroFilenameGenerator.h File Reference . . . . .	1493
11.95gdcmacroFileSet.h File Reference . . . . .	1494
11.96gdcmacroFileStreamer.h File Reference . . . . .	1495
11.97gdcmacroFindPatientRootQuery.h File Reference . . . . .	1496
11.98gdcmacroFindStudyRootQuery.h File Reference . . . . .	1497
11.99gdcmacroFragment.h File Reference . . . . .	1497
11.100gdcmacroGlobal.h File Reference . . . . .	1499
11.101gdcmacroGroupDict.h File Reference . . . . .	1500
11.102gdcmacroIconImage.h File Reference . . . . .	1500
11.103gdcmacroIconImageFilter.h File Reference . . . . .	1501
11.104gdcmacroIconImageGenerator.h File Reference . . . . .	1502
11.105gdcmacroImage.h File Reference . . . . .	1503
11.106gdcmacroImageApplyLookupTable.h File Reference . . . . .	1504
11.107gdcmacroImageChangePhotometricInterpretation.h File Reference . . . . .	1505



11.100	dcmImageChangePlanarConfiguration.h File Reference	1506
11.100	dcmImageChangeTransferSyntax.h File Reference	1506
11.110	dcmImageCodec.h File Reference	1507
11.110	dcmImageConverter.h File Reference	1508
11.110	dcmImageFragmentSplitter.h File Reference	1509
11.110	dcmImageHelper.h File Reference	1510
11.110	dcmImageReader.h File Reference	1511
11.115	dcmImageRegionReader.h File Reference	1512
11.110	dcmImageToImageFilter.h File Reference	1512
11.117	dcmImageWriter.h File Reference	1513
11.110	dcmImplementationClassUIDSub.h File Reference	1514
11.110	dcmImplementationUIDSub.h File Reference	1515
11.120	dcmImplementationVersionNameSub.h File Reference	1515
11.120	dcmImplicitDataElement.h File Reference	1517
11.120	dcmIOD.h File Reference	1517
11.120	dcmIODEntry.h File Reference	1519
11.120	dcmIODs.h File Reference	1521
11.125	dcmIPPSorter.h File Reference	1522
11.120	dcmItem.h File Reference	1523
11.127	dcmJPEG12Codec.h File Reference	1524
11.120	dcmJPEG16Codec.h File Reference	1525
11.120	dcmJPEG2000Codec.h File Reference	1525
11.130	dcmJPEG8Codec.h File Reference	1526
11.130	dcmJPEGCodec.h File Reference	1527
11.130	dcmJPEGLSCodec.h File Reference	1528
11.130	dcmJSON.h File Reference	1528
11.130	dcmKAKADUCodec.h File Reference	1529
11.135	dcmLegacyMacro.h File Reference	1530

11.135.1Macro Definition Documentation . . . . .	1530
11.135.1.1GDCM_LEGACY . . . . .	1531
11.135.1.2GDCM_LEGACY_BODY . . . . .	1531
11.135.1.3GDCM_LEGACY_REPLACED_BODY . . . . .	1531
11.136dcmLO.h File Reference . . . . .	1531
11.137dcmLookupTable.h File Reference . . . . .	1532
11.138dcmMacro.h File Reference . . . . .	1533
11.139dcmMacroEntry.h File Reference . . . . .	1535
11.139.1Macro Definition Documentation . . . . .	1536
11.139.1.1GDCMMACROENTRY_H . . . . .	1536
11.140dcmMacros.h File Reference . . . . .	1537
11.141dcmMaximumLengthSub.h File Reference . . . . .	1538
11.142dcmMD5.h File Reference . . . . .	1540
11.143dcmMediaStorage.h File Reference . . . . .	1540
11.144dcmMeshPrimitive.h File Reference . . . . .	1542
11.145dcmModalityPerformedProcedureStepCreateQuery.h File Reference . . . . .	1543
11.146dcmModalityPerformedProcedureStepSetQuery.h File Reference . . . . .	1544
11.147dcmModule.h File Reference . . . . .	1544
11.148dcmModuleEntry.h File Reference . . . . .	1546
11.149dcmModules.h File Reference . . . . .	1548
11.150dcmMovePatientRootQuery.h File Reference . . . . .	1549
11.151dcmMoveStudyRootQuery.h File Reference . . . . .	1550
11.152dcmMrProtocol.h File Reference . . . . .	1550
11.153dcmNActionMessages.h File Reference . . . . .	1552
11.154dcmNCreateMessages.h File Reference . . . . .	1553
11.155dcmNDeleteMessages.h File Reference . . . . .	1553
11.156dcmNestedModuleEntries.h File Reference . . . . .	1554
11.157dcmNetworkEvents.h File Reference . . . . .	1555

11.159dcmNetworkStateID.h File Reference . . . . .	1556
11.159dcmNEventReportMessages.h File Reference . . . . .	1557
11.160dcmNGetMessages.h File Reference . . . . .	1558
11.164dcmNormalizedMessageFactory.h File Reference . . . . .	1558
11.160dcmNormalizedNetworkFunctions.h File Reference . . . . .	1559
11.160dcmNSetMessages.h File Reference . . . . .	1560
11.164dcmObject.h File Reference . . . . .	1560
11.165dcmOpenSSLCryptoFactory.h File Reference . . . . .	1562
11.166dcmOpenSSLCryptographicMessageSyntax.h File Reference . . . . .	1562
11.167dcmOpenSSLP7CryptoFactory.h File Reference . . . . .	1564
11.168dcmOpenSSLP7CryptographicMessageSyntax.h File Reference . . . . .	1564
11.169dcmOrientation.h File Reference . . . . .	1566
11.170dcmOverlay.h File Reference . . . . .	1566
11.174dcmParseException.h File Reference . . . . .	1567
11.170dcmParser.h File Reference . . . . .	1569
11.170dcmPatient.h File Reference . . . . .	1569
11.174dcmPDataTFPDU.h File Reference . . . . .	1570
11.175dcmPDBElement.h File Reference . . . . .	1571
11.176dcmPDBHeader.h File Reference . . . . .	1572
11.177dcmPDFCodec.h File Reference . . . . .	1573
11.178dcmPDUFactory.h File Reference . . . . .	1574
11.178dcmPersonName.h File Reference . . . . .	1574
11.180dcmPGXCodec.h File Reference . . . . .	1575
11.184dcmPhotometricInterpretation.h File Reference . . . . .	1576
11.180dcmPixelFormat.h File Reference . . . . .	1577
11.180dcmPixmap.h File Reference . . . . .	1578
11.184dcmPixmapReader.h File Reference . . . . .	1579
11.185dcmPixmapToPixmapFilter.h File Reference . . . . .	1580

11.186dcmPixmapWriter.h File Reference . . . . .	1580
11.187dcmPNMCodec.h File Reference . . . . .	1582
11.188dcmPreamble.h File Reference . . . . .	1582
11.189dcmPresentationContext.h File Reference . . . . .	1584
11.190dcmPresentationContextAC.h File Reference . . . . .	1585
11.191dcmPresentationContextGenerator.h File Reference . . . . .	1586
11.192dcmPresentationContextRQ.h File Reference . . . . .	1586
11.193dcmPresentationDataValue.h File Reference . . . . .	1587
11.194dcmPrinter.h File Reference . . . . .	1588
11.195dcmPrivateTag.h File Reference . . . . .	1590
11.196dcmProgressEvent.h File Reference . . . . .	1591
11.197dcmPVRGCodec.h File Reference . . . . .	1592
11.198dcmPythonFilter.h File Reference . . . . .	1592
11.199dcmQueryBase.h File Reference . . . . .	1593
11.200dcmQueryFactory.h File Reference . . . . .	1594
11.201dcmQueryImage.h File Reference . . . . .	1595
11.202dcmQueryPatient.h File Reference . . . . .	1596
11.203dcmQuerySeries.h File Reference . . . . .	1597
11.204dcmQueryStudy.h File Reference . . . . .	1598
11.205dcmRAWCodec.h File Reference . . . . .	1599
11.206dcmReader.h File Reference . . . . .	1599
11.207dcmRegion.h File Reference . . . . .	1601
11.208dcmRescaler.h File Reference . . . . .	1602
11.209dcmRLECodec.h File Reference . . . . .	1603
11.210dcmRoleSelectionSub.h File Reference . . . . .	1603
11.211dcmScanner.h File Reference . . . . .	1604
11.212dcmSegment.h File Reference . . . . .	1605
11.213dcmSegmentedPaletteColorLookupTable.h File Reference . . . . .	1606

11.214dcmSegmentHelper.h File Reference . . . . .	1607
11.215dcmSegmentReader.h File Reference . . . . .	1608
11.216dcmSegmentWriter.h File Reference . . . . .	1609
11.217dcmSequenceOfFragments.h File Reference . . . . .	1610
11.218dcmSequenceOfItems.h File Reference . . . . .	1611
11.219dcmSerieHelper.h File Reference . . . . .	1611
11.220dcmSeries.h File Reference . . . . .	1613
11.221dcmServiceClassApplicationInformation.h File Reference . . . . .	1614
11.222dcmServiceClassUser.h File Reference . . . . .	1615
11.223dcmSHA1.h File Reference . . . . .	1615
11.224dcmSimpleSubjectWatcher.h File Reference . . . . .	1616
11.225dcmSmartPointer.h File Reference . . . . .	1617
11.226dcmSOPClassExtendedNegociationSub.h File Reference . . . . .	1618
11.227dcmSOPClassUIDToIOD.h File Reference . . . . .	1619
11.228dcmSorter.h File Reference . . . . .	1620
11.229dcmSpacing.h File Reference . . . . .	1622
11.230dcmSpectroscopy.h File Reference . . . . .	1622
11.231dcmSplitMosaicFilter.h File Reference . . . . .	1623
11.232dcmStaticAssert.h File Reference . . . . .	1623
11.232.1Macro Definition Documentation . . . . .	1624
11.232.1.1GDCM_DO_JOIN . . . . .	1624
11.232.1.2GDCM_DO_JOIN2 . . . . .	1624
11.232.1.3GDCM_JOIN . . . . .	1624
11.232.1.4GDCM_STATIC_ASSERT . . . . .	1625
11.233dcmStreamImageReader.h File Reference . . . . .	1625
11.234dcmStreamImageWriter.h File Reference . . . . .	1626
11.235dcmStrictScanner.h File Reference . . . . .	1626
11.236dcmString.h File Reference . . . . .	1627

11.237	dcmStringFilter.h File Reference . . . . .	1629
11.238	dcmStudy.h File Reference . . . . .	1629
11.239	dcmSubject.h File Reference . . . . .	1630
11.240	dcmSurface.h File Reference . . . . .	1631
11.241	dcmSurfaceHelper.h File Reference . . . . .	1632
11.242	dcmSurfaceReader.h File Reference . . . . .	1633
11.243	dcmSurfaceWriter.h File Reference . . . . .	1634
11.244	dcmSwapCode.h File Reference . . . . .	1634
11.245	dcmSwapper.h File Reference . . . . .	1635
11.246	dcmSystem.h File Reference . . . . .	1636
11.247	dcmTable.h File Reference . . . . .	1637
11.248	dcmTableEntry.h File Reference . . . . .	1638
11.249	dcmTableReader.h File Reference . . . . .	1640
11.250	dcmTag.h File Reference . . . . .	1641
11.251	dcmTagPath.h File Reference . . . . .	1642
11.252	dcmTagToVR.h File Reference . . . . .	1642
11.253	dcmTerminal.h File Reference . . . . .	1643
11.254	dcmTestDriver.h File Reference . . . . .	1644
11.255	dcmTesting.h File Reference . . . . .	1645
11.256	dcmTrace.h File Reference . . . . .	1646
11.256.1	Macro Definition Documentation . . . . .	1647
11.256.1.1	IGDCM_FUNCTION . . . . .	1647
11.256.1.2	dcmAssertAlwaysMacro . . . . .	1647
11.256.1.3	dcmAssertMacro . . . . .	1647
11.256.1.4	dcmDebugMacro . . . . .	1648
11.256.1.5	dcmErrorMacro . . . . .	1649
11.256.1.6	dcmWarningMacro . . . . .	1649
11.257	dcmTransferSyntax.h File Reference . . . . .	1650

11.250dcmTransferSyntaxSub.h File Reference . . . . .	1651
11.250dcmType.h File Reference . . . . .	1652
11.260dcmTypes.h File Reference . . . . .	1653
11.260dcmUIDGenerator.h File Reference . . . . .	1653
11.260dcmUIDs.h File Reference . . . . .	1654
11.260dcmULAction.h File Reference . . . . .	1655
11.260dcmULActionAA.h File Reference . . . . .	1656
11.260dcmULActionAE.h File Reference . . . . .	1656
11.260dcmULActionAR.h File Reference . . . . .	1657
11.260dcmULActionDT.h File Reference . . . . .	1658
11.260dcmULBasicCallback.h File Reference . . . . .	1658
11.260dcmULConnection.h File Reference . . . . .	1659
11.270dcmULConnectionCallback.h File Reference . . . . .	1660
11.270dcmULConnectionInfo.h File Reference . . . . .	1661
11.270dcmULConnectionManager.h File Reference . . . . .	1662
11.270dcmULEvent.h File Reference . . . . .	1663
11.270dcmULTransitionTable.h File Reference . . . . .	1664
11.270dcmULWritingCallback.h File Reference . . . . .	1665
11.270dcmUNExplicitDataElement.h File Reference . . . . .	1666
11.270dcmUNExplicitImplicitDataElement.h File Reference . . . . .	1666
11.270dcmUnpacker12Bits.h File Reference . . . . .	1667
11.270dcmUsage.h File Reference . . . . .	1667
11.280dcmUserInformation.h File Reference . . . . .	1669
11.280dcmUUIDGenerator.h File Reference . . . . .	1670
11.280dcmValidate.h File Reference . . . . .	1671
11.280dcmValue.h File Reference . . . . .	1671
11.280dcmValueIO.h File Reference . . . . .	1672
11.280dcmVersion.h File Reference . . . . .	1673

11.286dcmVL.h File Reference . . . . .	1674
11.287dcmVM.h File Reference . . . . .	1675
11.287.1Macro Definition Documentation . . . . .	1676
11.287.1.1TYPETOLENGTH . . . . .	1676
11.288dcmVR.h File Reference . . . . .	1676
11.288.1Macro Definition Documentation . . . . .	1678
11.288.1.1TYPETOENCODING . . . . .	1678
11.288.1.2VRTemplateCase . . . . .	1678
11.289dcmVR16ExplicitDataElement.h File Reference . . . . .	1678
11.290dcmWaveform.h File Reference . . . . .	1679
11.291dcmWin32.h File Reference . . . . .	1679
11.291.1Macro Definition Documentation . . . . .	1680
11.291.1.1GDCM_EXPORT . . . . .	1680
11.292dcmWLMFindQuery.h File Reference . . . . .	1680
11.293dcmWriter.h File Reference . . . . .	1681
11.294dcmXMLDictReader.h File Reference . . . . .	1682
11.295dcmXMLPrinter.h File Reference . . . . .	1682
11.296dcmXMLPrivateDictReader.h File Reference . . . . .	1683
11.297README.txt File Reference . . . . .	1684
11.298testsList.txt File Reference . . . . .	1684
11.299tkGDCMImageReader.h File Reference . . . . .	1684
11.299.1Macro Definition Documentation . . . . .	1685
11.299.1.1VTK_CMYK . . . . .	1685
11.299.1.2VTK_INVERSE_LUMINANCE . . . . .	1685
11.299.1.3VTK_LOOKUP_TABLE . . . . .	1685
11.299.1.4VTK_YBR . . . . .	1685
11.300tkGDCMImageReader2.h File Reference . . . . .	1686
11.300.1Macro Definition Documentation . . . . .	1686



11.300.1.1VTK_CMYK . . . . .	1686
11.300.1.2VTK_INVERSE_LUMINANCE . . . . .	1687
11.300.1.3VTK_LOOKUP_TABLE . . . . .	1687
11.300.1.4VTK_YBR . . . . .	1687
11.301tkGDCMImageWriter.h File Reference . . . . .	1687
11.302tkGDCMMedicalImageProperties.h File Reference . . . . .	1688
11.303tkGDCMPolyDataReader.h File Reference . . . . .	1688
11.304tkGDCMPolyDataWriter.h File Reference . . . . .	1689
11.305tkGDCMTesting.h File Reference . . . . .	1690
11.306tkGDCMThreadedImageReader.h File Reference . . . . .	1690
11.307tkGDCMThreadedImageReader2.h File Reference . . . . .	1691
11.308tkImageColorViewer.h File Reference . . . . .	1691
11.309tkImageMapToColors16.h File Reference . . . . .	1692
11.310tkImageMapToWindowLevelColors2.h File Reference . . . . .	1692
11.311tkImagePlanarComponentsToComponents.h File Reference . . . . .	1693
11.312tkImageRGBToYBR.h File Reference . . . . .	1693
11.313tkImageYBRToRGB.h File Reference . . . . .	1694
11.314tkLookupTable16.h File Reference . . . . .	1694
11.315tkRTStructSetProperties.h File Reference . . . . .	1695

<b>12 Example Documentation</b>	<b>1697</b>
12.1 AWTMedical3.java . . . . .	1697
12.2 BasicAnonymizer.cs . . . . .	1701
12.3 BasicImageAnonymizer.cs . . . . .	1702
12.4 CastConvertPhilips.py . . . . .	1704
12.5 ChangePrivateTags.cxx . . . . .	1706
12.6 ChangeSequenceUltrasound.cxx . . . . .	1707
12.7 CheckBigEndianBug.cxx . . . . .	1708
12.8 ClinicalTrialAnnotate.cxx . . . . .	1710
12.9 ClinicalTrialIdentificationWorkflow.cs . . . . .	1711
12.10CompressImage.cxx . . . . .	1714
12.11CompressLossyJPEG.cs . . . . .	1715
12.12Compute3DSpacing.cxx . . . . .	1716
12.13Convert16BitsTo8Bits.cxx . . . . .	1717
12.14ConvertMPL.py . . . . .	1718
12.15ConvertMultiFrameToSingleFrame.cxx . . . . .	1719
12.16ConvertNumpy.py . . . . .	1721
12.17ConvertPIL.py . . . . .	1721
12.18ConvertRGBToLuminance.cxx . . . . .	1722
12.19ConvertSingleBitTo8Bits.cxx . . . . .	1723
12.20ConvertToQImage.cxx . . . . .	1724
12.21CreateARGBImage.cxx . . . . .	1726
12.22CreateCMYKImage.cxx . . . . .	1727
12.23CreateFakePET.cxx . . . . .	1728
12.24CreateFakeRTDOSE.cxx . . . . .	1730
12.25CreateJPIPDataSet.cxx . . . . .	1732
12.26CreateRAWStorage.py . . . . .	1733
12.27csa2img.cxx . . . . .	1735

12.28CStoreQtProgress.cxx . . . . .	1737
12.29DecompressImage.cs . . . . .	1739
12.30DecompressImage.java . . . . .	1740
12.31DecompressImage.py . . . . .	1741
12.32DecompressImageMultiframe.cs . . . . .	1742
12.33DecompressJPEGFile.cs . . . . .	1744
12.34DecompressPixmap.java . . . . .	1745
12.35DeriveSeries.cxx . . . . .	1746
12.36DiffFile.cxx . . . . .	1747
12.37DiscriminateVolume.cxx . . . . .	1748
12.38DumbAnonymizer.py . . . . .	1752
12.39DumpADAC.cxx . . . . .	1754
12.40DumpCSA.cs . . . . .	1758
12.41DumpExamCard.cxx . . . . .	1759
12.42DumpGEMSMovieGroup.cxx . . . . .	1768
12.43DumpImageHeaderInfo.cxx . . . . .	1774
12.44DumpPhilipsECHO.cxx . . . . .	1776
12.45DumpSiemensBase64.cxx . . . . .	1782
12.46DumpToshibaDTI.cxx . . . . .	1783
12.47DumpToSQLITE3.cxx . . . . .	1784
12.48DuplicatePCDE.cxx . . . . .	1786
12.49ELSCINT1WaveToText.cxx . . . . .	1789
12.50EmptyMask.cxx . . . . .	1791
12.51EncapsulateFileInRawData.cxx . . . . .	1791
12.52ExtractEncapsulatedFile.cs . . . . .	1792
12.53ExtractEncryptedContent.cxx . . . . .	1794
12.54ExtractIconFromFile.cxx . . . . .	1795
12.55ExtractImageRegion.cs . . . . .	1796

12.56ExtractImageRegion.java . . . . .	1797
12.57ExtractImageRegionWithLUT.cs . . . . .	1798
12.58Extracting_All_Resolution.cxx . . . . .	1800
12.59ExtractOneFrame.cs . . . . .	1806
12.60Fake_Image_Using_Stream_Image_Writer.cxx . . . . .	1807
12.61FileAnonymize.cs . . . . .	1810
12.62FileAnonymize.java . . . . .	1811
12.63FileChangeTS.cs . . . . .	1811
12.64FileChangeTSLossy.cs . . . . .	1814
12.65FileStreaming.cs . . . . .	1816
12.66FindAllPatientName.py . . . . .	1817
12.67FixBrokenJ2K.cxx . . . . .	1818
12.68FixCommaBug.py . . . . .	1820
12.69FixJAIBugJPEGLS.cxx . . . . .	1820
12.70FixOrientation.cxx . . . . .	1823
12.71gdcmmorthoplanes.cxx . . . . .	1825
12.72gdcmmreslice.cxx . . . . .	1831
12.73gdcmmrtionplan.cxx . . . . .	1833
12.74gdcmmrtplan.cxx . . . . .	1837
12.75gdcmmscene.cxx . . . . .	1841
12.76gdcmmtexture.cxx . . . . .	1843
12.77gdcmmvolume.cxx . . . . .	1845
12.78GenAllIVR.cxx . . . . .	1846
12.79GenerateDICOMDIR.cs . . . . .	1848
12.80GenerateRTSTRUCT.cxx . . . . .	1849
12.81GenerateStandardSOPClasses.cxx . . . . .	1852
12.82GenFakeIdentifyFile.cxx . . . . .	1853
12.83GenFakeImage.cxx . . . . .	1855

12.84GenLongSeqs.cxx . . . . .	1857
12.85GenSeqs.cxx . . . . .	1859
12.86GetArray.cs . . . . .	1860
12.87GetJPEGSamplePrecision.cxx . . . . .	1861
12.88GetPortionCSAHeader.py . . . . .	1863
12.89GetSequenceUltrasound.cxx . . . . .	1864
12.90GetSubSequenceData.cxx . . . . .	1865
12.91headsq2dcm.py . . . . .	1868
12.92HelloActiviz.cs . . . . .	1869
12.93HelloActiviz2.cs . . . . .	1870
12.94HelloActiviz3.cs . . . . .	1871
12.95HelloActiviz4.cs . . . . .	1872
12.96HelloActiviz5.cs . . . . .	1873
12.97HelloSimple.java . . . . .	1874
12.98HelloVizWorld.cxx . . . . .	1875
12.99HelloVTKWorld.cs . . . . .	1876
12.100HelloVTKWorld.java . . . . .	1877
12.101HelloVTKWorld2.cs . . . . .	1878
12.102HelloWorld.cxx . . . . .	1879
12.103HelloWorld.py . . . . .	1880
12.104J22tomultisc.cxx . . . . .	1880
12.105LargeVRDSExplicit.cxx . . . . .	1882
12.106MagnifyFile.cxx . . . . .	1884
12.107MakeTemplate.cxx . . . . .	1885
12.108ManipulateFile.cs . . . . .	1886
12.109ManipulateFile.py . . . . .	1887
12.110ManipulateSequence.py . . . . .	1888
12.111MergeFile.py . . . . .	1889

12.112MergeTwoFiles.cxx . . . . .	1890
12.113MetalImageMD5Activiz.cs . . . . .	1891
12.114MIPViewer.java . . . . .	1893
12.115mpegVideoInfo.cs . . . . .	1895
12.116MPRViewer.java . . . . .	1899
12.117MPRViewer2.java . . . . .	1901
12.118MrProtocol.cxx . . . . .	1906
12.119NewSequence.cs . . . . .	1913
12.120NewSequence.py . . . . .	1914
12.121offscreenimage.cxx . . . . .	1915
12.122PatchFile.cxx . . . . .	1916
12.123PhilipsPrivateRescaleInterceptSlope.py . . . . .	1918
12.124PlaySound.py . . . . .	1918
12.125pmsct_rgb1.cxx . . . . .	1919
12.126PrivateDict.py . . . . .	1923
12.127PublicDict.cxx . . . . .	1923
12.128QIDO-RS.cxx . . . . .	1924
12.129ReadAndDumpDICOMDIR.cxx . . . . .	1925
12.130ReadAndDumpDICOMDIR.py . . . . .	1928
12.131ReadAndDumpDICOMDIR2.cxx . . . . .	1930
12.132ReadAndPrintAttributes.cxx . . . . .	1935
12.133ReadExplicitLengthSQIVR.cxx . . . . .	1937
12.134ReadFiles.java . . . . .	1938
12.135ReadGEMSSDO.cxx . . . . .	1939
12.136ReadMultiTimesException.cxx . . . . .	1941
12.137ReadSeriesIntoVTK.java . . . . .	1942
12.138ReadUTF8QtDir.cxx . . . . .	1943
12.139RefCounting.cs . . . . .	1945

12.140	<a href="#">ReformatFile.cs</a>	1946
12.141	<a href="#">RemovePrivateTags.py</a>	1947
12.142	<a href="#">RescaleImage.cs</a>	1947
12.143	<a href="#">Reslicesphere.cxx</a>	1948
12.144	<a href="#">ReWriteSCAsMR.py</a>	1956
12.145	<a href="#">Re2img.cxx</a>	1957
12.146	<a href="#">structapp.cxx</a>	1960
12.147	<a href="#">ScanDirectory.cs</a>	1962
12.148	<a href="#">ScanDirectory.java</a>	1963
12.149	<a href="#">ScanDirectory.py</a>	1966
12.150	<a href="#">SendFileSCU.cs</a>	1967
12.151	<a href="#">SimplePrint.cs</a>	1968
12.152	<a href="#">SimplePrintPatientName.cs</a>	1969
12.153	<a href="#">SimpleScanner.cxx</a>	1970
12.154	<a href="#">SortImage.cxx</a>	1971
12.155	<a href="#">SortImage.py</a>	1973
12.156	<a href="#">SortImage2.cs</a>	1974
12.157	<a href="#">StandardizeFiles.cs</a>	1974
12.158	<a href="#">StreamImageReaderTest.cxx</a>	1976
12.159	<a href="#">TemplateEmptyImage.cxx</a>	1979
12.160	<a href="#">TestByteSwap.cxx</a>	1981
12.161	<a href="#">TestReader.cxx</a>	1983
12.162	<a href="#">TestReader.py</a>	1984
12.163	<a href="#">Threadgdcm.cxx</a>	1985
12.164	<a href="#">TraverseModules.cxx</a>	1988
12.165	<a href="#">uid_unique.cxx</a>	1989
12.166	<a href="#">VolumeSorter.cxx</a>	1990
12.167	<a href="#">WriteBuffer.py</a>	1993





## Chapter 1

# GDCM Documentation

This is the developpers documentation.

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

<http://gdcm.sourceforge.net/2.8/gdcm-2.8.6.pdf>

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

<http://gdcm.sourceforge.net/2.8/gdcm-2.8.6-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

### Class `gdcm::IPPSorter`

There are currently a couple of bugs in this implementation:





## Chapter 5

# Namespace Index

### 5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">gdc</a>	43
<a href="#">gdc::network</a>	77
<a href="#">gdc::SegmentHelper</a>	84
<a href="#">gdc::terminal</a>	
Class for Terminal	84



## Chapter 6

# Hierarchical Index

### 6.1 Class Hierarchy

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

gdcmm::network::AbstractSyntax . . . . .	104
gdcmm::network::ApplicationContext . . . . .	119
gdcmm::ApplicationEntity . . . . .	121
gdcmm::network::ARTIMTimer . . . . .	129
gdcmm::ASN1 . . . . .	130
gdcmm::network::AsynchronousOperationsWindowSub . . . . .	132
gdcmm::Attribute< Group, Element, TVR, TVM > . . . . .	134
gdcmm::Attribute< Group, Element, TVR, VM::VM1 > . . . . .	142
gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > . . . . .	152
gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . . .	149
gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . . .	150
gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > . . . . .	160
gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . . .	158
gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > . . . . .	163
gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . . .	161
gdcmm::Base64 . . . . .	167
gdcmm::network::BaseCompositeMessage . . . . .	169
gdcmm::network::CEchoRQ . . . . .	234
gdcmm::network::CEchoRSP . . . . .	236
gdcmm::network::CFindCancelRQ . . . . .	238
gdcmm::network::CFindRQ . . . . .	239
gdcmm::network::CFindRSP . . . . .	241
gdcmm::network::CMoveCancelRq . . . . .	242
gdcmm::network::CMoveRQ . . . . .	243
gdcmm::network::CMoveRSP . . . . .	245
gdcmm::network::CStoreRQ . . . . .	297
gdcmm::network::CStoreRSP . . . . .	298
gdcmm::network::BaseNormalizedMessage . . . . .	171
gdcmm::network::NActionRQ . . . . .	712
gdcmm::network::NActionRSP . . . . .	713

gdcmm::network::NCreateRQ . . . . .	715
gdcmm::network::NCreateRSP . . . . .	716
gdcmm::network::NDeleteRQ . . . . .	718
gdcmm::network::NDeleteRSP . . . . .	719
gdcmm::network::NEventReportRQ . . . . .	724
gdcmm::network::NEventReportRSP . . . . .	725
gdcmm::network::NGetRQ . . . . .	727
gdcmm::network::NGetRSP . . . . .	728
gdcmm::network::NSetRQ . . . . .	735
gdcmm::network::NSetRSP . . . . .	737
gdcmm::network::BasePDU . . . . .	173
gdcmm::network::AAabortPDU . . . . .	87
gdcmm::network::AAAssociateACPDU . . . . .	90
gdcmm::network::AAAssociateRJPDU . . . . .	94
gdcmm::network::AAAssociateRQPDU . . . . .	97
gdcmm::network::AReleaseRPPDU . . . . .	124
gdcmm::network::AReleaseRQPDU . . . . .	126
gdcmm::network::PDataTFPDU . . . . .	773
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar > . . . . .	1032
gdcmm::SegmentHelper::BasicCodedEntry . . . . .	187
gdcmm::BitmapToBitmapFilter . . . . .	207
gdcmm::PixmapToPixmapFilter . . . . .	817
gdcmm::ImageToImageFilter . . . . .	572
gdcmm::ImageApplyLookupTable . . . . .	523
gdcmm::ImageChangePhotometricInterpretation . . . . .	526
gdcmm::ImageChangePlanarConfiguration . . . . .	530
gdcmm::ImageChangeTransferSyntax . . . . .	534
gdcmm::ImageFragmentSplitter . . . . .	554
gdcmm::ByteBuffer . . . . .	215
gdcmm::ByteSwap< T > . . . . .	217
gdcmm::ByteSwapFilter . . . . .	219
gdcmm::network::CFind . . . . .	237
gdcmm::Coder . . . . .	247
gdcmm::Codec . . . . .	246
gdcmm::AudioCodec . . . . .	164
gdcmm::ImageCodec . . . . .	539
gdcmm::DeltaEncodingCodec . . . . .	349
gdcmm::JPEG2000Codec . . . . .	613
gdcmm::JPEGCodec . . . . .	623
gdcmm::JPEG12Codec . . . . .	606
gdcmm::JPEG16Codec . . . . .	610
gdcmm::JPEG8Codec . . . . .	620
gdcmm::JPEGLSCCodec . . . . .	631
gdcmm::KAKADUCodec . . . . .	641
gdcmm::PGXCodec . . . . .	792
gdcmm::PNMCodec . . . . .	825
gdcmm::PVRGCodec . . . . .	866
gdcmm::RAWCodec . . . . .	886
gdcmm::RLECodec . . . . .	906
gdcmm::PDFCodec . . . . .	783
gdcmm::CodeString . . . . .	249

gdcm::network::CompositeMessageFactory . . . . .	259
gdcm::CompositeNetworkFunctions . . . . .	261
gdcm::ConstCharWrapper . . . . .	266
gdcm::CryptoFactory . . . . .	269
gdcm::CAPICryptoFactory . . . . .	229
gdcm::OpenSSLCryptoFactory . . . . .	742
gdcm::OpenSSLP7CryptoFactory . . . . .	747
gdcm::CryptographicMessageSyntax . . . . .	272
gdcm::CAPICryptographicMessageSyntax . . . . .	231
gdcm::OpenSSLCryptographicMessageSyntax . . . . .	744
gdcm::OpenSSLP7CryptographicMessageSyntax . . . . .	749
gdcm::CSAElement . . . . .	275
gdcm::CSAHeader . . . . .	283
gdcm::CSAHeaderDict . . . . .	290
gdcm::CSAHeaderDictEntry . . . . .	293
gdcm::DataElement . . . . .	306
gdcm::CP246ExplicitDataElement . . . . .	267
gdcm::ExplicitDataElement . . . . .	433
gdcm::ExplicitImplicitDataElement . . . . .	436
gdcm::Fragment . . . . .	498
gdcm::BasicOffsetTable . . . . .	190
gdcm::ImplicitDataElement . . . . .	583
gdcm::Item . . . . .	601
gdcm::UNExplicitDataElement . . . . .	1227
gdcm::UNExplicitImplicitDataElement . . . . .	1230
gdcm::VR16ExplicitDataElement . . . . .	1268
gdcm::DataSet . . . . .	325
gdcm::CommandDataSet . . . . .	257
gdcm::FileMetaInformation . . . . .	462
gdcm::DataSetHelper . . . . .	341
gdcm::Decoder . . . . .	342
gdcm::Codec . . . . .	246
gdcm::DefinedTerms . . . . .	344
gdcm::Defs . . . . .	344
gdcm::DICOMDIR . . . . .	351
gdcm::DICOMDIRGenerator . . . . .	352
gdcm::Dict . . . . .	356
gdcm::DictConverter . . . . .	360
gdcm::DictEntry . . . . .	365
gdcm::Dicts . . . . .	372
gdcm::network::DIMSE . . . . .	376
gdcm::DirectionCosines . . . . .	378
gdcm::Directory . . . . .	381
gdcm::DirectoryHelper . . . . .	386
gdcm::DummyValueGenerator . . . . .	388
gdcm::Element< TVR, TVM > . . . . .	391
gdcm::Element< TVR, VM::VM1_n > . . . . .	398
gdcm::Element< TVR, VM::VM1_2 > . . . . .	396
gdcm::Element< TVR, VM::VM2_n > . . . . .	405
gdcm::Element< TVR, VM::VM2_2n > . . . . .	403
gdcm::Element< TVR, VM::VM3_n > . . . . .	409
gdcm::Element< TVR, VM::VM3_3n > . . . . .	407

gdcmm::Element< VR::AS, VM::VM5 > . . . . .	411
gdcmm::Element< VR::OB, VM::VM1_n > . . . . .	391
gdcmm::Element< VR::OB, VM::VM1 > . . . . .	412
gdcmm::Element< VR::OW, VM::VM1_n > . . . . .	391
gdcmm::Element< VR::OW, VM::VM1 > . . . . .	413
gdcmm::ElementDisableCombinations< TVR, TVM > . . . . .	415
gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . . .	416
gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . . .	416
gdcmm::EmptyMaskGenerator . . . . .	416
gdcmm::EncapsulatedDocument . . . . .	419
gdcmm::EncodingImplementation< T > . . . . .	420
gdcmm::EncodingImplementation< VR::VRASCII > . . . . .	421
gdcmm::EncodingImplementation< VR::VRBINARY > . . . . .	423
gdcmm::EnumeratedValues . . . . .	425
gdcmm::Event . . . . .	426
gdcmm::AnyEvent . . . . .	118
gdcmm::AbortEvent . . . . .	103
gdcmm::AnonymizeEvent . . . . .	107
gdcmm::DataEvent . . . . .	321
gdcmm::DataSetEvent . . . . .	338
gdcmm::EndEvent . . . . .	424
gdcmm::ExitEvent . . . . .	432
gdcmm::FileNameEvent . . . . .	475
gdcmm::InitializeEvent . . . . .	586
gdcmm::IterationEvent . . . . .	605
gdcmm::ModifiedEvent . . . . .	692
gdcmm::ProgressEvent . . . . .	862
gdcmm::StartEvent . . . . .	1009
gdcmm::UserEvent . . . . .	1236
gdcmm::NoEvent . . . . .	730
std::exception	
gdcmm::CSAHeaderDictException . . . . .	296
gdcmm::DataElementException . . . . .	321
gdcmm::Exception . . . . .	430
gdcmm::ParseException . . . . .	766
gdcmm::Fiducials . . . . .	439
gdcmm::FileDerivation . . . . .	455
gdcmm::FileExplicitFilter . . . . .	459
gdcmm::Filename . . . . .	471
gdcmm::FilenameGenerator . . . . .	479
gdcmm::FileSet . . . . .	483
gdcmm::Global . . . . .	501
gdcmm::GroupDict . . . . .	505
gdcmm::IconImageFilter . . . . .	508
gdcmm::IconImageGenerator . . . . .	511
gdcmm::ignore_char . . . . .	515
gdcmm::ImageConverter . . . . .	553
gdcmm::ImageHelper . . . . .	557
gdcmm::network::ImplementationClassUIDSub . . . . .	578
gdcmm::network::ImplementationUIDSub . . . . .	580
gdcmm::network::ImplementationVersionNameSub . . . . .	581
gdcmm::IOD . . . . .	587
gdcmm::IODEntry . . . . .	590

gdcmm::IODs	593
gdcmm::JSON	638
gdcmm::Scanner::Itstr	655
gdcmm::StrictScanner::Itstr	656
gdcmm::Macro	656
gdcmm::Macros	660
gdcmm::network::MaximumLengthSub	662
gdcmm::MD5	664
gdcmm::MediaStorage	665
gdcmm::Module	693
gdcmm::ModuleEntry	696
gdcmm::NestedModuleEntries	721
gdcmm::Modules	701
gdcmm::MrProtocol	709
gdcmm::network::NormalizedMessageFactory	731
gdcmm::NormalizedNetworkFunctions	732
gdcmm::Object	738
gdcmm::BaseQuery	176
gdcmm::BaseRootQuery	182
gdcmm::FindPatientRootQuery	492
gdcmm::FindStudyRootQuery	495
gdcmm::MovePatientRootQuery	703
gdcmm::MoveStudyRootQuery	706
gdcmm::WLMFindQuery	1390
gdcmm::ModalityPerformedProcedureStepCreateQuery	686
gdcmm::ModalityPerformedProcedureStepSetQuery	689
gdcmm::Bitmap	192
gdcmm::Pixmap	807
gdcmm::Image	516
gdcmm::Curve	300
gdcmm::File	439
gdcmm::FileWithName	490
gdcmm::LookupTable	648
gdcmm::SegmentedPaletteColorLookupTable	935
gdcmm::MeshPrimitive	680
gdcmm::Overlay	756
gdcmm::Segment	925
gdcmm::Subject	1042
gdcmm::Anonymizer	110
gdcmm::Command	254
gdcmm::MemberCommand< T >	675
gdcmm::SimpleMemberCommand< T >	980
gdcmm::FileAnonymizer	444
gdcmm::FileChangeTransferSyntax	448
gdcmm::FileDecompressLookupTable	452
gdcmm::FileStreamer	485
gdcmm::network::ULConnectionManager	1213
gdcmm::Scanner	915
gdcmm::ServiceClassUser	970
gdcmm::StrictScanner	1023
gdcmm::Surface	1045
gdcmm::Value	1244
gdcmm::ByteValue	221

gdcmm::SequenceOfFragments . . . . .	946
gdcmm::SequenceOfItems . . . . .	953
gdcmm::Orientation . . . . .	752
gdcmm::Parser . . . . .	768
gdcmm::Patient . . . . .	772
gdcmm::PDBelement . . . . .	776
gdcmm::PDBHeader . . . . .	779
gdcmm::network::PDUFactory . . . . .	785
gdcmm::PersonName . . . . .	789
gdcmm::PhotometricInterpretation . . . . .	795
gdcmm::PixelFormat . . . . .	798
gdcmm::Preamble . . . . .	828
gdcmm::PresentationContext . . . . .	833
gdcmm::network::PresentationContextAC . . . . .	837
gdcmm::PresentationContextGenerator . . . . .	839
gdcmm::network::PresentationContextRQ . . . . .	843
gdcmm::network::PresentationDataValue . . . . .	847
gdcmm::Printer . . . . .	851
gdcmm::DictPrinter . . . . .	370
gdcmm::Dumper . . . . .	389
gdcmm::PrivateDict . . . . .	856
gdcmm::PythonFilter . . . . .	869
gdcmm::QueryBase . . . . .	871
gdcmm::QueryImage . . . . .	876
gdcmm::QueryPatient . . . . .	878
gdcmm::QuerySeries . . . . .	881
gdcmm::QueryStudy . . . . .	883
gdcmm::QueryFactory . . . . .	874
gdcmm::Reader . . . . .	890
gdcmm::PixmapReader . . . . .	813
gdcmm::ImageReader . . . . .	564
gdcmm::ImageRegionReader . . . . .	568
gdcmm::SegmentReader . . . . .	938
gdcmm::SurfaceReader . . . . .	1063
gdcmm::RealWorldValueMappingContent . . . . .	897
gdcmm::Region . . . . .	899
gdcmm::BoxRegion . . . . .	210
gdcmm::Rescaler . . . . .	901
gdcmm::network::RoleSelectionSub . . . . .	912
gdcmm::SerieHelper::Rule . . . . .	914
gdcmm::SerieHelper . . . . .	963
gdcmm::Series . . . . .	968
gdcmm::network::ServiceClassApplicationInformation . . . . .	969
gdcmm::SHA1 . . . . .	978
gdcmm::SimpleSubjectWatcher . . . . .	984
gdcmm::MrProtocol::Slice . . . . .	988
gdcmm::MrProtocol::SliceArray . . . . .	989
gdcmm::SmartPointer< ObjectType > . . . . .	990
gdcmm::SmartPointer< gdcmm::Bitmap > . . . . .	990
gdcmm::SmartPointer< gdcmm::File > . . . . .	990
gdcmm::SmartPointer< gdcmm::Image > . . . . .	990
gdcmm::SmartPointer< gdcmm::MemberCommand > . . . . .	990
gdcmm::SmartPointer< gdcmm::MeshPrimitive > . . . . .	990



gdcm::SmartPointer< gdcm::Pixmap > . . . . .	990
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . . .	990
gdcm::SmartPointer< gdcm::Subject > . . . . .	990
gdcm::SmartPointer< LookupTable > . . . . .	990
gdcm::SmartPointer< Segment > . . . . .	990
gdcm::SmartPointer< Surface > . . . . .	990
gdcm::SmartPointer< Value > . . . . .	990
gdcm::network::SOPClassExtendedNegociationSub . . . . .	994
gdcm::SOPClassUIDToIOD . . . . .	996
gdcm::Sorter . . . . .	998
gdcm::IPPSorter . . . . .	596
gdcm::Spacing . . . . .	1003
gdcm::Spectroscopy . . . . .	1005
gdcm::SplitMosaicFilter . . . . .	1006
gdcm::static_assert_test< x > . . . . .	1010
gdcm::STATIC_ASSERTION_FAILURE< x > . . . . .	1010
gdcm::STATIC_ASSERTION_FAILURE< true > . . . . .	1011
gdcm::StreamImageReader . . . . .	1011
gdcm::StreamImageWriter . . . . .	1015
String<"\", 64 > . . . . .	
gdcm::LO . . . . .	644
gdcm::StringFilter . . . . .	1037
gdcm::Study . . . . .	1041
gdcm::SurfaceHelper . . . . .	1060
gdcm::SwapCode . . . . .	1070
gdcm::SwapperDoOp . . . . .	1072
gdcm::SwapperNoOp . . . . .	1073
gdcm::System . . . . .	1074
gdcm::Table . . . . .	1081
gdcm::TableEntry . . . . .	1083
gdcm::TableReader . . . . .	1084
gdcm::XMLDictReader . . . . .	1400
gdcm::XMLPrivateDictReader . . . . .	1407
gdcm::network::TableRow . . . . .	1089
gdcm::Tag . . . . .	1090
gdcm::PrivateTag . . . . .	859
gdcm::TagPath . . . . .	1101
gdcm::Testing . . . . .	1104
gdcm::Trace . . . . .	1112
gdcm::TransferSyntax . . . . .	1117
gdcm::network::TransferSyntaxSub . . . . .	1124
gdcm::network::Transition . . . . .	1126
gdcm::Type . . . . .	1128
gdcm::UI . . . . .	1131
gdcm::UIDGenerator . . . . .	1132
gdcm::UIDs . . . . .	1134
gdcm::network::ULAction . . . . .	1160
gdcm::network::ULActionAA1 . . . . .	1163
gdcm::network::ULActionAA2 . . . . .	1165
gdcm::network::ULActionAA3 . . . . .	1166
gdcm::network::ULActionAA4 . . . . .	1167
gdcm::network::ULActionAA5 . . . . .	1169
gdcm::network::ULActionAA6 . . . . .	1170

gdcmm::network::ULActionAA7	.1171
gdcmm::network::ULActionAA8	.1173
gdcmm::network::ULActionAE1	.1174
gdcmm::network::ULActionAE2	.1175
gdcmm::network::ULActionAE3	.1177
gdcmm::network::ULActionAE4	.1178
gdcmm::network::ULActionAE5	.1179
gdcmm::network::ULActionAE6	.1181
gdcmm::network::ULActionAE7	.1182
gdcmm::network::ULActionAE8	.1183
gdcmm::network::ULActionAR1	.1185
gdcmm::network::ULActionAR10	.1186
gdcmm::network::ULActionAR2	.1187
gdcmm::network::ULActionAR3	.1189
gdcmm::network::ULActionAR4	.1190
gdcmm::network::ULActionAR5	.1191
gdcmm::network::ULActionAR6	.1193
gdcmm::network::ULActionAR7	.1194
gdcmm::network::ULActionAR8	.1195
gdcmm::network::ULActionAR9	.1197
gdcmm::network::ULActionDT1	.1198
gdcmm::network::ULActionDT2	.1199
gdcmm::network::ULConnection	.1203
gdcmm::network::ULConnectionCallback	.1208
gdcmm::network::ULBasicCallback	.1201
gdcmm::network::ULWritingCallback	.1225
gdcmm::network::ULConnectionInfo	.1211
gdcmm::network::ULEvent	.1221
gdcmm::network::ULTransitionTable	.1223
gdcmm::Unpacker12Bits	.1232
gdcmm::Usage	.1233
gdcmm::network::UserInformation	.1237
gdcmm::UUIDGenerator	.1240
gdcmm::Validate	.1241
gdcmm::ValueIO< TDE, TSwap, TType >	.1247
gdcmm::MrProtocol::Vector3	.1248
gdcmm::Version	.1248
gdcmm::VL	.1251
gdcmm::VM	.1255
gdcmm::VMToLength< T >	.1260
gdcmm::VR	.1260
gdcmm::VRToEncoding< T >	.1270
gdcmm::VRToType< T >	.1270
gdcmm::VRToType< TVR >	.1270
gdcmm::VRVLSIZE< T >	.1271
gdcmm::VRVLSIZE< 0 >	.1271
gdcmm::VRVLSIZE< 1 >	.1272
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	.1367
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	.1363
vtkImageWriter	
vtkGDCMImageWriter	.1302
vtkLookupTable	

vtkLookupTable16 . . . . .	.1374
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties . . . . .	.1312
vtkMedicalImageReader2	
vtkGDCMImageReader . . . . .	.1273
vtkGDCMThreadedImageReader . . . . .	.1329
vtkGDCMImageReader2 . . . . .	.1288
vtkObject	
vtkGDCMTesting . . . . .	.1325
vtkImageColorViewer . . . . .	.1342
vtkRTStructSetProperties . . . . .	.1378
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader . . . . .	.1315
vtkPolyDataWriter	
vtkGDCMPolyDataWriter . . . . .	.1320
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2 . . . . .	.1333
vtkImageMapToColors16 . . . . .	.1357
vtkImageRGBToYBR . . . . .	.1369
vtkImageYBRToRGB . . . . .	.1372
gdcm::Waveform . . . . .	.1389
gdcm::Writer . . . . .	.1393
gdcm::PixmapWriter . . . . .	.820
gdcm::ImageWriter . . . . .	.575
gdcm::SegmentWriter . . . . .	.942
gdcm::SurfaceWriter . . . . .	.1067
gdcm::XMLPrinter . . . . .	.1403



## Chapter 7

# Class Index

### 7.1 Class List

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

gdcn::network::AAabortPDU	
AAabortPDU	87
gdcn::network::AAAssociateACPDU	
AAAssociateACPDU	90
gdcn::network::AAAssociateRJPDU	
AAAssociateRJPDU	94
gdcn::network::AAAssociateRQPDU	
AAAssociateRQPDU	97
gdcn::AbortEvent	103
gdcn::network::AbstractSyntax	
AbstractSyntax	104
gdcn::AnonymizeEvent	
AnonymizeEvent	107
gdcn::Anonymizer	
Anonymizer	110
gdcn::AnyEvent	118
gdcn::network::ApplicationContext	
ApplicationContext	119
gdcn::ApplicationEntity	
ApplicationEntity	121
gdcn::network::AReleaseRPPDU	
AReleaseRPPDU	124
gdcn::network::AReleaseRQPDU	
AReleaseRQPDU	126
gdcn::network::ARTIMTimer	
ARTIMTimer	129
gdcn::ASN1	
Class for ASN1	130
gdcn::network::AsynchronousOperationsWindowSub	
AsynchronousOperationsWindowSub	132

<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, TVM &gt;</a>	
<a href="#">Attribute</a> class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary	134
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1 &gt;</a>	142
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_3 &gt;</a>	149
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_8 &gt;</a>	150
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM1_n &gt;</a>	152
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM2_2n &gt;</a>	158
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM2_n &gt;</a>	160
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM3_3n &gt;</a>	161
<a href="#">gdcmm::Attribute&lt; Group, Element, TVR, VM::VM3_n &gt;</a>	163
<a href="#">gdcmm::AudioCodec</a>	
<a href="#">AudioCodec</a>	164
<a href="#">gdcmm::Base64</a>	
Class for <a href="#">Base64</a>	167
<a href="#">gdcmm::network::BaseCompositeMessage</a>	
<a href="#">BaseCompositeMessage</a>	169
<a href="#">gdcmm::network::BaseNormalizedMessage</a>	
<a href="#">BaseNormalizedMessage</a>	171
<a href="#">gdcmm::network::BasePDU</a>	
<a href="#">BasePDU</a>	173
<a href="#">gdcmm::BaseQuery</a>	
<a href="#">BaseQuery</a>	176
<a href="#">gdcmm::BaseRootQuery</a>	
<a href="#">BaseRootQuery</a>	182
<a href="#">gdcmm::SegmentHelper::BasicCodedEntry</a>	
This structure defines a basic coded entry with all of its attributes	187
<a href="#">gdcmm::BasicOffsetTable</a>	
Class to represent a <a href="#">BasicOffsetTable</a>	190
<a href="#">gdcmm::Bitmap</a>	
<a href="#">Bitmap</a> class	192
<a href="#">gdcmm::BitmapToBitmapFilter</a>	
<a href="#">BitmapToBitmapFilter</a> class	207
<a href="#">gdcmm::BoxRegion</a>	
Class for manipulation box region	210
<a href="#">gdcmm::ByteBuffer</a>	
<a href="#">ByteBuffer</a>	215
<a href="#">gdcmm::ByteSwap&lt; T &gt;</a>	
<a href="#">ByteSwap</a>	217
<a href="#">gdcmm::ByteSwapFilter</a>	
<a href="#">ByteSwapFilter</a>	219
<a href="#">gdcmm::ByteValue</a>	
Class to represent binary value (array of bytes)	221
<a href="#">gdcmm::CAPICryptoFactory</a>	229
<a href="#">gdcmm::CAPICryptographicMessageSyntax</a>	231
<a href="#">gdcmm::network::CEchoRQ</a>	
<a href="#">CEchoRQ</a>	234
<a href="#">gdcmm::network::CEchoRSP</a>	
<a href="#">CEchoRSP</a> this file defines the messages for the cecho action	236
<a href="#">gdcmm::network::CFind</a>	237
<a href="#">gdcmm::network::CFindCancelRQ</a>	
<a href="#">CFindCancelRQ</a> this file defines the messages for the cfind action	238
<a href="#">gdcmm::network::CFindRQ</a>	
<a href="#">CFindRQ</a>	239

gdcm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	241
gdcm::network::CMoveCancelRq	242
gdcm::network::CMoveRQ	
CMoveRQ	243
gdcm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	245
gdcm::Codec	
Codec class	246
gdcm::Coder	
Coder	247
gdcm::CodeString	
CodeString	249
gdcm::Command	
Command superclass for callback/observer methods	254
gdcm::CommandDataSet	
Class to represent a <a href="#">Command DataSet</a>	257
gdcm::network::CompositeMessageFactory	
CompositeMessageFactory	259
gdcm::CompositeNetworkFunctions	
Composite Network Functions	261
gdcm::ConstCharWrapper	
Do not use me	266
gdcm::CP246ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as CP246Explicit Data <a href="#">Element</a>	267
gdcm::CryptoFactory	
Class to do handle the crypto factory	269
gdcm::CryptographicMessageSyntax	272
gdcm::CSAElement	
Class to represent a CSA <a href="#">Element</a>	275
gdcm::CSAHeader	
Class for <a href="#">CSAHeader</a>	283
gdcm::CSAHeaderDict	
Class to represent a map of <a href="#">CSAHeaderDictEntry</a>	290
gdcm::CSAHeaderDictEntry	
Class to represent an Entry in the <a href="#">Dict</a>	293
gdcm::CSAHeaderDictException	296
gdcm::network::CStoreRQ	
CStoreRQ	297
gdcm::network::CStoreRSP	
CStoreRSP this file defines the messages for the cecho action	298
gdcm::Curve	
Curve class to handle element 50xx,3000 <a href="#">Curve</a> Data	300
gdcm::DataElement	
Class to represent a Data <a href="#">Element</a> either Implicit or Explicit	306
gdcm::DataElementException	321
gdcm::DataEvent	
DataEvent	321
gdcm::DataSet	
Class to represent a Data Set (which contains Data Elements)	325
gdcm::DataSetEvent	
DataSetEvent	338
gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	341

gdcmm::Decoder	
Decoder	342
gdcmm::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	344
gdcmm::Defs	
FIXME I do not like the name 'Defs'	344
gdcmm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	349
gdcmm::DICOMDIR	
DICOMDIR class	351
gdcmm::DICOMDIRGenerator	
DICOMDIRGenerator class	352
gdcmm::Dict	
Class to represent a map of DictEntry	356
gdcmm::DictConverter	
Class to convert a .dic file into something else:	360
gdcmm::DictEntry	
Class to represent an Entry in the Dict	365
gdcmm::DictPrinter	
DictPrinter class	370
gdcmm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	372
gdcmm::network::DIMSE	
DIMSE	376
gdcmm::DirectionCosines	
Class to handle DirectionCosines	378
gdcmm::Directory	
Class for manipulation directories	381
gdcmm::DirectoryHelper	
DirectoryHelper	386
gdcmm::DummyValueGenerator	
Class for generating dummy value	388
gdcmm::Dumper	
Codec class	389
gdcmm::Element< TVR, TVM >	
Element class	391
gdcmm::Element< TVR, VM::VM1_2 >	396
gdcmm::Element< TVR, VM::VM1_n >	398
gdcmm::Element< TVR, VM::VM2_2n >	403
gdcmm::Element< TVR, VM::VM2_n >	405
gdcmm::Element< TVR, VM::VM3_3n >	407
gdcmm::Element< TVR, VM::VM3_n >	409
gdcmm::Element< VR::AS, VM::VM5 >	411
gdcmm::Element< VR::OB, VM::VM1 >	412
gdcmm::Element< VR::OW, VM::VM1 >	413



gdcm::ElementDisableCombinations< TVR, TVM >	
A class which is used to produce compile errors for an invalid combination of template parameters	415
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	416
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	416
gdcm::EmptyMaskGenerator	
EmptyMaskGenerator Main class to generate a Empty Mask <a href="#">Series</a> from an input <a href="#">Series</a> . This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM <a href="#">Series</a> within the same input directory	416
gdcm::EncapsulatedDocument	
EncapsulatedDocument	419
gdcm::EncodingImplementation< T >	
EncodingImplementation	420
gdcm::EncodingImplementation< VR::VRASCII >	421
gdcm::EncodingImplementation< VR::VRBINARY >	423
gdcm::EndEvent	424
gdcm::EnumeratedValues	
Element. A Data <a href="#">Element</a> with Enumerated Values that does not have a <a href="#">Value</a> 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:	425
gdcm::Event	
Superclass for callback/observer methods	426
gdcm::Exception	
Exception	430
gdcm::ExitEvent	432
gdcm::ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a>	433
gdcm::ExplicitImplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a>	436
gdcm::Fiducials	
Fiducials	439
gdcm::File	
DICOM File	439
gdcm::FileAnonymizer	
FileAnonymizer	444
gdcm::FileChangeTransferSyntax	
FileChangeTransferSyntax	448
gdcm::FileDecompressLookupTable	
FileDecompressLookupTable class	452
gdcm::FileDerivation	
FileDerivation class	455
gdcm::FileExplicitFilter	
FileExplicitFilter class	459
gdcm::FileMetaInformation	
Class to represent a <a href="#">File</a> Meta Information	462
gdcm::Filename	
Class to manipulate file name's	471
gdcm::FileNameEvent	
FileNameEvent	475
gdcm::FilenameGenerator	
FilenameGenerator	479
gdcm::FileSet	483
gdcm::FileStreamer	
FileStreamer	485

gdcm::FileWithName	
FileWithName	490
gdcm::FindPatientRootQuery	
PatientRootQuery	492
gdcm::FindStudyRootQuery	
FindStudyRootQuery	495
gdcm::Fragment	
Class to represent a <a href="#">Fragment</a>	498
gdcm::Global	
Global	501
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	505
gdcm::IconImageFilter	
IconImageFilter	508
gdcm::IconImageGenerator	
IconImageGenerator	511
gdcm::ignore_char	515
gdcm::Image	
Image	516
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class	523
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class	526
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class	530
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class	534
gdcm::ImageCodec	
ImageCodec	539
gdcm::ImageConverter	
Image Converter	553
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class	554
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	557
gdcm::ImageReader	
ImageReader	564
gdcm::ImageRegionReader	
ImageRegionReader	568
gdcm::ImageToImageFilter	
ImageToImageFilter class	572
gdcm::ImageWriter	
ImageWriter	575
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub	578
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub	580
gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub	581
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data <a href="#">Element</a>	583
gdcm::InitializeEvent	586
gdcm::IOD	
Class for representing a <a href="#">IOD</a>	587

<a href="#">gdcm::IODEntry</a>	
Class for representing a <a href="#">IODEntry</a>	590
<a href="#">gdcm::IODs</a>	
Class for representing a <a href="#">IODs</a>	593
<a href="#">gdcm::IPPSorter</a>	
IPPSorter	596
<a href="#">gdcm::Item</a>	
Class to represent an <a href="#">Item</a>	601
<a href="#">gdcm::IterationEvent</a>	605
<a href="#">gdcm::JPEG12Codec</a>	
Class to do JPEG 12bits (lossy & lossless)	606
<a href="#">gdcm::JPEG16Codec</a>	
Class to do JPEG 16bits (lossless)	610
<a href="#">gdcm::JPEG2000Codec</a>	
Class to do JPEG 2000	613
<a href="#">gdcm::JPEG8Codec</a>	
Class to do JPEG 8bits (lossy & lossless)	620
<a href="#">gdcm::JPEGCodec</a>	
JPEG codec	623
<a href="#">gdcm::JPEGLSCodec</a>	
JPEG-LS	631
<a href="#">gdcm::JSON</a>	638
<a href="#">gdcm::KAKADUCodec</a>	
KAKADUCodec	641
<a href="#">gdcm::LO</a>	
LO	644
<a href="#">gdcm::LookupTable</a>	
LookupTable class	648
<a href="#">gdcm::Scanner::Itstr</a>	655
<a href="#">gdcm::StrictScanner::Itstr</a>	656
<a href="#">gdcm::Macro</a>	
Class for representing a <a href="#">Macro</a>	656
<a href="#">gdcm::Macros</a>	
Class for representing a <a href="#">Modules</a>	660
<a href="#">gdcm::network::MaximumLengthSub</a>	
MaximumLengthSub	662
<a href="#">gdcm::MD5</a>	
Class for <a href="#">MD5</a>	664
<a href="#">gdcm::MediaStorage</a>	
MediaStorage	665
<a href="#">gdcm::MemberCommand&lt; T &gt;</a>	
Command subclass that calls a pointer to a member function	675
<a href="#">gdcm::MeshPrimitive</a>	
This class defines surface mesh primitives	680
<a href="#">gdcm::ModalityPerformedProcedureStepCreateQuery</a>	
ModalityPerformedProcedureStepCreateQuery	686
<a href="#">gdcm::ModalityPerformedProcedureStepSetQuery</a>	
ModalityPerformedProcedureStepSetQuery	689
<a href="#">gdcm::ModifiedEvent</a>	692
<a href="#">gdcm::Module</a>	
Class for representing a <a href="#">Module</a>	693
<a href="#">gdcm::ModuleEntry</a>	
Class for representing a <a href="#">ModuleEntry</a>	696

gdcmm::Modules	
Class for representing a <a href="#">Modules</a>	701
gdcmm::MovePatientRootQuery	
MovePatientRootQuery	703
gdcmm::MoveStudyRootQuery	
MoveStudyRootQuery	706
gdcmm::MrProtocol	
Class for <a href="#">MrProtocol</a>	709
gdcmm::network::NActionRQ	
NActionRQ	712
gdcmm::network::NActionRSP	
NActionRSP this file defines the messages for the NAction action	713
gdcmm::network::NCreateRQ	
NCreateRQ	715
gdcmm::network::NCreateRSP	
NCreateRSP this file defines the messages for the ncreate action	716
gdcmm::network::NDeleteRQ	
NDeleteRQ	718
gdcmm::network::NDeleteRSP	
NDeleteRSP this file defines the messages for the ndelete action	719
gdcmm::NestedModuleEntries	
Class for representing a <a href="#">NestedModuleEntries</a>	721
gdcmm::network::NEventReportRQ	
NEventReportRQ	724
gdcmm::network::NEventReportRSP	
NEventReportRSP this file defines the messages for the neventreport action	725
gdcmm::network::NGetRQ	
NGetRQ	727
gdcmm::network::NGetRSP	
NGetRSP this file defines the messages for the nget action	728
gdcmm::NoEvent	730
gdcmm::network::NormalizedMessageFactory	731
gdcmm::NormalizedNetworkFunctions	
Normalized Network Functions	732
gdcmm::network::NSetRQ	
NSetRQ	735
gdcmm::network::NSetRSP	
NSetRSP this file defines the messages for the nset action	737
gdcmm::Object	
Object	738
gdcmm::OpenSSLCryptoFactory	742
gdcmm::OpenSSLCryptographicMessageSyntax	744
gdcmm::OpenSSLP7CryptoFactory	747
gdcmm::OpenSSLP7CryptographicMessageSyntax	749
gdcmm::Orientation	
Class to handle <a href="#">Orientation</a>	752
gdcmm::Overlay	
Overlay class	756
gdcmm::ParseException	
ParseException Standard exception handling object	766
gdcmm::Parser	
Parser ala XML_Parser from expat (SAX)	768
gdcmm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	772

gdcmm::network::PDataTFPDU	
PDataTFPDU	773
gdcmm::PDBElement	
Class to represent a PDB Element	776
gdcmm::PDBHeader	
Class for PDBHeader	779
gdcmm::PDFCodec	
PDFCodec class	783
gdcmm::network::PDUFactory	
PDUFactory basically, given an initial byte, construct the	785
gdcmm::PersonName	
PersonName class	789
gdcmm::PGXCodec	
Class to do PGX	792
gdcmm::PhotometricInterpretation	
Class to represent an PhotometricInterpretation	795
gdcmm::PixelFormat	
PixelFormat	798
gdcmm::Pixmap	
Pixmap class	807
gdcmm::PixmapReader	
PixmapReader	813
gdcmm::PixmapToPixmapFilter	
PixmapToPixmapFilter class	817
gdcmm::PixmapWriter	
PixmapWriter	820
gdcmm::PNMCodec	
Class to do PNM	825
gdcmm::Preamble	
DICOM Preamble (Part 10)	828
gdcmm::PresentationContext	
PresentationContext	833
gdcmm::network::PresentationContextAC	
PresentationContextAC	837
gdcmm::PresentationContextGenerator	
PresentationContextGenerator	839
gdcmm::network::PresentationContextRQ	
PresentationContextRQ	843
gdcmm::network::PresentationDataValue	
PresentationDataValue	847
gdcmm::Printer	
Printer class	851
gdcmm::PrivateDict	
Private Dict	856
gdcmm::PrivateTag	
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)	859
gdcmm::ProgressEvent	
ProgressEvent	862
gdcmm::PVRGCodec	
PVRGCodec	866
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	869

gdcm::QueryBase	
QueryBase	871
gdcm::QueryFactory	
QueryFactory.h	874
gdcm::QueryImage	
QueryImage	876
gdcm::QueryPatient	
QueryPatient	878
gdcm::QuerySeries	
QuerySeries	881
gdcm::QueryStudy	
QueryStudy.h	883
gdcm::RAWCodec	
RAWCodec class	886
gdcm::Reader	
Reader ala DOM (Document Object Model)	890
gdcm::RealWorldValueMappingContent	897
gdcm::Region	
Class for manipulation region	899
gdcm::Rescaler	
Rescale class	901
gdcm::RLECodec	
Class to do RLE	906
gdcm::network::RoleSelectionSub	
RoleSelectionSub	912
gdcm::SerieHelper::Rule	914
gdcm::Scanner	
Scanner	915
gdcm::Segment	
This class defines a segment	925
gdcm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable class	935
gdcm::SegmentReader	
This class defines a segment reader	938
gdcm::SegmentWriter	
This class defines a segment writer	942
gdcm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	946
gdcm::SequenceOfItems	
Class to represent a Sequence Of Items	953
gdcm::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 disapear soon, you've been warned	963
gdcm::Series	
Series	968
gdcm::network::ServiceClassApplicationInformation	969
gdcm::ServiceClassUser	
ServiceClassUser	970
gdcm::SHA1	
Class for SHA1	978
gdcm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function	980
gdcm::SimpleSubjectWatcher	
SimpleSubjectWatcher	984

gdcm::MrProtocol::Slice	988
gdcm::MrProtocol::SliceArray	989
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	990
gdcm::network::SOPClassExtendedNegociationSub	
SOPClassExtendedNegociationSub	994
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	996
gdcm::Sorter	
Sorter	998
gdcm::Spacing	
Class for Spacing	1003
gdcm::Spectroscopy	
Spectroscopy class	1005
gdcm::SplitMosaicFilter	
SplitMosaicFilter class	1006
gdcm::StartEvent	1009
gdcm::static_assert_test< x >	1010
gdcm::STATIC_ASSERTION_FAILURE< x >	1010
gdcm::STATIC_ASSERTION_FAILURE< true >	1011
gdcm::StreamImageReader	
StreamImageReader	1011
gdcm::StreamImageWriter	
StreamImageReader	1015
gdcm::StrictScanner	
StrictScanner	1023
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	1032
gdcm::StringFilter	
StringFilter	1037
gdcm::Study	
Study	1041
gdcm::Subject	
Subject	1042
gdcm::Surface	
This class defines a SURFACE IE	1045
gdcm::SurfaceHelper	
SurfaceHelper	1060
gdcm::SurfaceReader	
This class defines a SURFACE IE reader	1063
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer	1067
gdcm::SwapCode	
SwapCode representation	1070
gdcm::SwapperDoOp	1072
gdcm::SwapperNoOp	1073
gdcm::System	
Class to do system operation	1074
gdcm::Table	
Table	1081
gdcm::TableEntry	
TableEntry	1083
gdcm::TableReader	
Class for representing a TableReader	1084

<a href="#">gdcmm::network::TableRow</a>	1089
<a href="#">gdcmm::Tag</a>	
Class to represent a DICOM Data <a href="#">Element</a> ( <a href="#">Attribute</a> ) <a href="#">Tag</a> (Group, <a href="#">Element</a> )	1090
<a href="#">gdcmm::TagPath</a>	
Class to handle a path of tag	1101
<a href="#">gdcmm::Testing</a>	
Class for testing	1104
<a href="#">gdcmm::Trace</a>	
Trace	1112
<a href="#">gdcmm::TransferSyntax</a>	
Class to manipulate Transfer Syntax	1117
<a href="#">gdcmm::network::TransferSyntaxSub</a>	
TransferSyntaxSub	1124
<a href="#">gdcmm::network::Transition</a>	1126
<a href="#">gdcmm::Type</a>	
Type	1128
<a href="#">gdcmm::UI</a>	1131
<a href="#">gdcmm::UIDGenerator</a>	
Class for generating unique UID	1132
<a href="#">gdcmm::UIDs</a>	
All known uids	1134
<a href="#">gdcmm::network::ULAction</a>	
ULAction	1160
<a href="#">gdcmm::network::ULActionAA1</a>	1163
<a href="#">gdcmm::network::ULActionAA2</a>	1165
<a href="#">gdcmm::network::ULActionAA3</a>	1166
<a href="#">gdcmm::network::ULActionAA4</a>	1167
<a href="#">gdcmm::network::ULActionAA5</a>	1169
<a href="#">gdcmm::network::ULActionAA6</a>	1170
<a href="#">gdcmm::network::ULActionAA7</a>	1171
<a href="#">gdcmm::network::ULActionAA8</a>	1173
<a href="#">gdcmm::network::ULActionAE1</a>	1174
<a href="#">gdcmm::network::ULActionAE2</a>	1175
<a href="#">gdcmm::network::ULActionAE3</a>	1177
<a href="#">gdcmm::network::ULActionAE4</a>	1178
<a href="#">gdcmm::network::ULActionAE5</a>	1179
<a href="#">gdcmm::network::ULActionAE6</a>	1181
<a href="#">gdcmm::network::ULActionAE7</a>	1182
<a href="#">gdcmm::network::ULActionAE8</a>	1183
<a href="#">gdcmm::network::ULActionAR1</a>	1185
<a href="#">gdcmm::network::ULActionAR10</a>	1186
<a href="#">gdcmm::network::ULActionAR2</a>	1187
<a href="#">gdcmm::network::ULActionAR3</a>	1189
<a href="#">gdcmm::network::ULActionAR4</a>	1190
<a href="#">gdcmm::network::ULActionAR5</a>	1191
<a href="#">gdcmm::network::ULActionAR6</a>	1193
<a href="#">gdcmm::network::ULActionAR7</a>	1194
<a href="#">gdcmm::network::ULActionAR8</a>	1195
<a href="#">gdcmm::network::ULActionAR9</a>	1197
<a href="#">gdcmm::network::ULActionDT1</a>	1198
<a href="#">gdcmm::network::ULActionDT2</a>	1199
<a href="#">gdcmm::network::ULBasicCallback</a>	
ULBasicCallback	1201



gdcm::network::ULConnection	
ULConnection	1203
gdcm::network::ULConnectionCallback	1208
gdcm::network::ULConnectionInfo	
ULConnectionInfo	1211
gdcm::network::ULConnectionManager	
ULConnectionManager	1213
gdcm::network::ULEvent	
ULEvent	1221
gdcm::network::ULTransitionTable	
ULTransitionTable	The transition table of all the ULEvents, new ULActions, and ULStates . . . . . 1223
gdcm::network::ULWritingCallback	1225
gdcm::UNExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as UNExplicit Data <a href="#">Element</a>	1227
gdcm::UNExplicitImplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a>	1230
gdcm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	1232
gdcm::Usage	
Usage	1233
gdcm::UserEvent	1236
gdcm::network::UserInformation	
UserInformation	1237
gdcm::UUIDGenerator	
Class for generating unique UUID	1240
gdcm::Validate	
Validate class	1241
gdcm::Value	
Class to represent the value of a Data <a href="#">Element</a>	1244
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	1247
gdcm::MrProtocol::Vector3	1248
gdcm::Version	
Major/minor and build version	1248
gdcm::VL	
Value Length	1251
gdcm::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	1255
gdcm::VMToLength< T >	1260
gdcm::VR	
VR class	1260
gdcm::VR16ExplicitDataElement	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a>	1268
gdcm::VRToEncoding< T >	1270
gdcm::VRToType< T >	1270
gdcm::VRVLSize< T >	1271
gdcm::VRVLSize< 0 >	1271
gdcm::VRVLSize< 1 >	1272
vtkGDCMImageReader	1273
vtkGDCMImageReader2	1288
vtkGDCMImageWriter	1302
vtkGDCMMedicalImageProperties	1312
vtkGDCMPolyDataReader	1315

<a href="#">vtkGDCMPolyDataWriter</a>	1320
<a href="#">vtkGDCMTesting</a>	1325
<a href="#">vtkGDCMThreadedImageReader</a>	1329
<a href="#">vtkGDCMThreadedImageReader2</a>	1333
<a href="#">vtkImageColorViewer</a>	1342
<a href="#">vtkImageMapToColors16</a>	1357
<a href="#">vtkImageMapToWindowLevelColors2</a>	1363
<a href="#">vtkImagePlanarComponentsToComponents</a>	1367
<a href="#">vtkImageRGBToYBR</a>	1369
<a href="#">vtkImageYBRToRGB</a>	1372
<a href="#">vtkLookupTable16</a>	1374
<a href="#">vtkRTStructSetProperties</a>	1378
<a href="#">gdcm::Waveform</a>	
<a href="#">Waveform</a> class	1389
<a href="#">gdcm::WLMFindQuery</a>	
<a href="#">PatientRootQuery</a>	1390
<a href="#">gdcm::Writer</a>	
<a href="#">Writer</a> ala DOM (Document <a href="#">Object</a> Model)	1393
<a href="#">gdcm::XMLDictReader</a>	
Class for representing a <a href="#">XMLDictReader</a>	1400
<a href="#">gdcm::XMLPrinter</a>	1403
<a href="#">gdcm::XMLPrivateDictReader</a>	
Class for representing a <a href="#">XMLPrivateDictReader</a>	1407

## Chapter 8

# File Index

### 8.1 File List

Here is a list of all files with brief descriptions:

<a href="#">gdcmAAabortPDU.h</a>	1411
<a href="#">gdcmAAAssociateACPDU.h</a>	1412
<a href="#">gdcmAAAssociateRJPDU.h</a>	1412
<a href="#">gdcmAAAssociateRQPDU.h</a>	1413
<a href="#">gdcmAbstractSyntax.h</a>	1414
<a href="#">gdcmAnonymizeEvent.h</a>	1415
<a href="#">gdcmAnonymizer.h</a>	1416
<a href="#">gdcmApplicationContext.h</a>	1417
<a href="#">gdcmApplicationEntity.h</a>	1418
<a href="#">gdcmAReleaseRPPDU.h</a>	1418
<a href="#">gdcmAReleaseRQPDU.h</a>	1419
<a href="#">gdcmARTIMTimer.h</a>	1420
<a href="#">gdcmASN1.h</a>	1421
<a href="#">gdcmAsynchronousOperationsWindowSub.h</a>	1422
<a href="#">gdcmAttribute.h</a>	1422
<a href="#">gdcmAudioCodec.h</a>	1424
<a href="#">gdcmBase64.h</a>	1424
<a href="#">gdcmBaseCompositeMessage.h</a>	1425
<a href="#">gdcmBaseNormalizedMessage.h</a>	1426
<a href="#">gdcmBasePDU.h</a>	1427
<a href="#">gdcmBaseQuery.h</a>	1428
<a href="#">gdcmBaseRootQuery.h</a>	1429
<a href="#">gdcmBasicOffsetTable.h</a>	1430
<a href="#">gdcmBitmap.h</a>	1431
<a href="#">gdcmBitmapToBitmapFilter.h</a>	1432
<a href="#">gdcmBoxRegion.h</a>	1433
<a href="#">gdcmByteBuffer.h</a>	1433
<a href="#">gdcmByteSwap.h</a>	1435
<a href="#">gdcmByteSwapFilter.h</a>	1435
<a href="#">gdcmByteValue.h</a>	1436
<a href="#">gdcmCAPICryptoFactory.h</a>	1437

<a href="#">gdcmCAPICryptographicMessageSyntax.h</a>	1438
<a href="#">gdcmCEchoMessages.h</a>	1438
<a href="#">gdcmCFindMessages.h</a>	1439
<a href="#">gdcmCMoveMessages.h</a>	1440
<a href="#">gdcmCodec.h</a>	1441
<a href="#">gdcmCoder.h</a>	1442
<a href="#">gdcmCodeString.h</a>	1444
<a href="#">gdcmCommand.h</a>	1445
<a href="#">gdcmCommandDataSet.h</a>	1446
<a href="#">gdcmCompositeMessageFactory.h</a>	1447
<a href="#">gdcmCompositeNetworkFunctions.h</a>	1447
<a href="#">gdcmConstCharWrapper.h</a>	1448
<a href="#">gdcmCP246ExplicitDataElement.h</a>	1448
<a href="#">gdcmCryptoFactory.h</a>	1449
<a href="#">gdcmCryptographicMessageSyntax.h</a>	1450
<a href="#">gdcmCSAElement.h</a>	1451
<a href="#">gdcmCSAHeader.h</a>	1452
<a href="#">gdcmCSAHeaderDict.h</a>	1453
<a href="#">gdcmCSAHeaderDictEntry.h</a>	1454
<a href="#">gdcmCStoreMessages.h</a>	1455
<a href="#">gdcmCurve.h</a>	1456
<a href="#">gdcmDataElement.h</a>	1457
<a href="#">gdcmDataEvent.h</a>	1459
<a href="#">gdcmDataSet.h</a>	1460
<a href="#">gdcmDataSetEvent.h</a>	1461
<a href="#">gdcmDataSetHelper.h</a>	1461
<a href="#">gdcmDecoder.h</a>	1462
<a href="#">gdcmDefinedTerms.h</a>	1464
<a href="#">gdcmDeflateStream.h</a>	1464
<a href="#">gdcmDefs.h</a>	1465
<a href="#">gdcmDeltaEncodingCodec.h</a>	1466
<a href="#">gdcmDICOmdir.h</a>	1466
<a href="#">gdcmDICOmdirGenerator.h</a>	1467
<a href="#">gdcmDict.h</a>	1468
<a href="#">gdcmDictConverter.h</a>	1469
<a href="#">gdcmDictEntry.h</a>	1470
<a href="#">gdcmDictPrinter.h</a>	1471
<a href="#">gdcmDicts.h</a>	1472
<a href="#">gdcmDIMSE.h</a>	1473
<a href="#">gdcmDirectionCosines.h</a>	1473
<a href="#">gdcmDirectory.h</a>	1474
<a href="#">gdcmDirectoryHelper.h</a>	1475
<a href="#">gdcmDummyValueGenerator.h</a>	1476
<a href="#">gdcmDumper.h</a>	1476
<a href="#">gdcmElement.h</a>	1477
<a href="#">gdcmEmptyMaskGenerator.h</a>	1479
<a href="#">gdcmEncapsulatedDocument.h</a>	1480
<a href="#">gdcmEnumeratedValues.h</a>	1481
<a href="#">gdcmEvent.h</a>	1481
<a href="#">gdcmException.h</a>	1483
<a href="#">gdcmExplicitDataElement.h</a>	1484
<a href="#">gdcmExplicitImplicitDataElement.h</a>	1485
<a href="#">gdcmFiducials.h</a>	1485
<a href="#">gdcmFile.h</a>	1486

<a href="#">gdcmFileAnonymizer.h</a>	1487
<a href="#">gdcmFileChangeTransferSyntax.h</a>	1488
<a href="#">gdcmFileDecompressLookupTable.h</a>	1488
<a href="#">gdcmFileDerivation.h</a>	1489
<a href="#">gdcmFileExplicitFilter.h</a>	1490
<a href="#">gdcmFileMetaInformation.h</a>	1491
<a href="#">gdcmFilename.h</a>	1492
<a href="#">gdcmFileNameEvent.h</a>	1492
<a href="#">gdcmFilenameGenerator.h</a>	1493
<a href="#">gdcmFileSet.h</a>	1494
<a href="#">gdcmFileStreamer.h</a>	1495
<a href="#">gdcmFindPatientRootQuery.h</a>	1496
<a href="#">gdcmFindStudyRootQuery.h</a>	1497
<a href="#">gdcmFragment.h</a>	1497
<a href="#">gdcmGlobal.h</a>	1499
<a href="#">gdcmGroupDict.h</a>	1500
<a href="#">gdcmIconImage.h</a>	1500
<a href="#">gdcmIconImageFilter.h</a>	1501
<a href="#">gdcmIconImageGenerator.h</a>	1502
<a href="#">gdcmImage.h</a>	1503
<a href="#">gdcmImageApplyLookupTable.h</a>	1504
<a href="#">gdcmImageChangePhotometricInterpretation.h</a>	1505
<a href="#">gdcmImageChangePlanarConfiguration.h</a>	1506
<a href="#">gdcmImageChangeTransferSyntax.h</a>	1506
<a href="#">gdcmImageCodec.h</a>	1507
<a href="#">gdcmImageConverter.h</a>	1508
<a href="#">gdcmImageFragmentSplitter.h</a>	1509
<a href="#">gdcmImageHelper.h</a>	1510
<a href="#">gdcmImageReader.h</a>	1511
<a href="#">gdcmImageRegionReader.h</a>	1512
<a href="#">gdcmImageToImageFilter.h</a>	1512
<a href="#">gdcmImageWriter.h</a>	1513
<a href="#">gdcmImplementationClassUIDSub.h</a>	1514
<a href="#">gdcmImplementationUIDSub.h</a>	1515
<a href="#">gdcmImplementationVersionNameSub.h</a>	1515
<a href="#">gdcmImplicitDataElement.h</a>	1517
<a href="#">gdcmIOD.h</a>	1517
<a href="#">gdcmIODEntry.h</a>	1519
<a href="#">gdcmIODs.h</a>	1521
<a href="#">gdcmIPPSorter.h</a>	1522
<a href="#">gdcmItem.h</a>	1523
<a href="#">gdcmJPEG12Codec.h</a>	1524
<a href="#">gdcmJPEG16Codec.h</a>	1525
<a href="#">gdcmJPEG2000Codec.h</a>	1525
<a href="#">gdcmJPEG8Codec.h</a>	1526
<a href="#">gdcmJPEGCodec.h</a>	1527
<a href="#">gdcmJPEGLSCodec.h</a>	1528
<a href="#">gdcmJSON.h</a>	1528
<a href="#">gdcmKAKADUCodec.h</a>	1529
<a href="#">gdcmLegacyMacro.h</a>	1530
<a href="#">gdcmLO.h</a>	1531
<a href="#">gdcmLookupTable.h</a>	1532
<a href="#">gdcmMacro.h</a>	1533
<a href="#">gdcmMacroEntry.h</a>	1535

gdcmMacros.h	1537
gdcmMaximumLengthSub.h	1538
gdcmMD5.h	1540
gdcmMediaStorage.h	1540
gdcmMeshPrimitive.h	1542
gdcmModalityPerformedProcedureStepCreateQuery.h	1543
gdcmModalityPerformedProcedureStepSetQuery.h	1544
gdcmModule.h	1544
gdcmModuleEntry.h	1546
gdcmModules.h	1548
gdcmMovePatientRootQuery.h	1549
gdcmMoveStudyRootQuery.h	1550
gdcmMrProtocol.h	1550
gdcmNActionMessages.h	1552
gdcmNCreateMessages.h	1553
gdcmNDeleteMessages.h	1553
gdcmNestedModuleEntries.h	1554
gdcmNetworkEvents.h	1555
gdcmNetworkStateID.h	1556
gdcmNEventReportMessages.h	1557
gdcmNGetMessages.h	1558
gdcmNormalizedMessageFactory.h	1558
gdcmNormalizedNetworkFunctions.h	1559
gdcmNSetMessages.h	1560
gdcmObject.h	1560
gdcmOpenSSLCryptoFactory.h	1562
gdcmOpenSSLCryptographicMessageSyntax.h	1562
gdcmOpenSSLP7CryptoFactory.h	1564
gdcmOpenSSLP7CryptographicMessageSyntax.h	1564
gdcmOrientation.h	1566
gdcmOverlay.h	1566
gdcmParseException.h	1567
gdcmParser.h	1569
gdcmPatient.h	1569
gdcmPDataTFPDU.h	1570
gdcmPDBelement.h	1571
gdcmPDBHeader.h	1572
gdcmPDFCodec.h	1573
gdcmPDUFactory.h	1574
gdcmPersonName.h	1574
gdcmPGXCodec.h	1575
gdcmPhotometricInterpretation.h	1576
gdcmPixelFormat.h	1577
gdcmPixmap.h	1578
gdcmPixmapReader.h	1579
gdcmPixmapToPixmapFilter.h	1580
gdcmPixmapWriter.h	1580
gdcmPNMCodec.h	1582
gdcmPreamble.h	1582
gdcmPresentationContext.h	1584
gdcmPresentationContextAC.h	1585
gdcmPresentationContextGenerator.h	1586
gdcmPresentationContextRQ.h	1586
gdcmPresentationDataValue.h	1587

gdcmPrinter.h	1588
gdcmPrivateTag.h	1590
gdcmProgressEvent.h	1591
gdcmPVRGCodec.h	1592
gdcmPythonFilter.h	1592
gdcmQueryBase.h	1593
gdcmQueryFactory.h	1594
gdcmQueryImage.h	1595
gdcmQueryPatient.h	1596
gdcmQuerySeries.h	1597
gdcmQueryStudy.h	1598
gdcmRAWCodec.h	1599
gdcmReader.h	1599
gdcmRegion.h	1601
gdcmRescaler.h	1602
gdcmRLECodec.h	1603
gdcmRoleSelectionSub.h	1603
gdcmScanner.h	1604
gdcmSegment.h	1605
gdcmSegmentedPaletteColorLookupTable.h	1606
gdcmSegmentHelper.h	1607
gdcmSegmentReader.h	1608
gdcmSegmentWriter.h	1609
gdcmSequenceOfFragments.h	1610
gdcmSequenceOfItems.h	1611
gdcmSerieHelper.h	1611
gdcmSeries.h	1613
gdcmServiceClassApplicationInformation.h	1614
gdcmServiceClassUser.h	1615
gdcmSHA1.h	1615
gdcmSimpleSubjectWatcher.h	1616
gdcmSmartPointer.h	1617
gdcmSOPClassExtendedNegociationSub.h	1618
gdcmSOPClassUIDToIOD.h	1619
gdcmSorter.h	1620
gdcmSpacing.h	1622
gdcmSpectroscopy.h	1622
gdcmSplitMosaicFilter.h	1623
gdcmStaticAssert.h	1623
gdcmStreamImageReader.h	1625
gdcmStreamImageWriter.h	1626
gdcmStrictScanner.h	1626
gdcmString.h	1627
gdcmStringFilter.h	1629
gdcmStudy.h	1629
gdcmSubject.h	1630
gdcmSurface.h	1631
gdcmSurfaceHelper.h	1632
gdcmSurfaceReader.h	1633
gdcmSurfaceWriter.h	1634
gdcmSwapCode.h	1634
gdcmSwapper.h	1635
gdcmSystem.h	1636
gdcmTable.h	1637

<a href="#">gdcmTableEntry.h</a>	1638
<a href="#">gdcmTableReader.h</a>	1640
<a href="#">gdcmTag.h</a>	1641
<a href="#">gdcmTagPath.h</a>	1642
<a href="#">gdcmTagToVR.h</a>	1642
<a href="#">gdcmTerminal.h</a>	1643
<a href="#">gdcmTestDriver.h</a>	1644
<a href="#">gdcmTesting.h</a>	1645
<a href="#">gdcmTrace.h</a>	1646
<a href="#">gdcmTransferSyntax.h</a>	1650
<a href="#">gdcmTransferSyntaxSub.h</a>	1651
<a href="#">gdcmType.h</a>	1652
<a href="#">gdcmTypes.h</a>	1653
<a href="#">gdcmUIDGenerator.h</a>	1653
<a href="#">gdcmUIDs.h</a>	1654
<a href="#">gdcmULAction.h</a>	1655
<a href="#">gdcmULActionAA.h</a>	1656
<a href="#">gdcmULActionAE.h</a>	1656
<a href="#">gdcmULActionAR.h</a>	1657
<a href="#">gdcmULActionDT.h</a>	1658
<a href="#">gdcmULBasicCallback.h</a>	1658
<a href="#">gdcmULConnection.h</a>	1659
<a href="#">gdcmULConnectionCallback.h</a>	1660
<a href="#">gdcmULConnectionInfo.h</a>	1661
<a href="#">gdcmULConnectionManager.h</a>	1662
<a href="#">gdcmULEvent.h</a>	1663
<a href="#">gdcmULTransitionTable.h</a>	1664
<a href="#">gdcmULWritingCallback.h</a>	1665
<a href="#">gdcmUNExplicitDataElement.h</a>	1666
<a href="#">gdcmUNExplicitImplicitDataElement.h</a>	1666
<a href="#">gdcmUnpacker12Bits.h</a>	1667
<a href="#">gdcmUsage.h</a>	1667
<a href="#">gdcmUserInformation.h</a>	1669
<a href="#">gdcmUUIDGenerator.h</a>	1670
<a href="#">gdcmValidate.h</a>	1671
<a href="#">gdcmValue.h</a>	1671
<a href="#">gdcmValueIO.h</a>	1672
<a href="#">gdcmVersion.h</a>	1673
<a href="#">gdcmVL.h</a>	1674
<a href="#">gdcmVM.h</a>	1675
<a href="#">gdcmVR.h</a>	1676
<a href="#">gdcmVR16ExplicitDataElement.h</a>	1678
<a href="#">gdcmWaveform.h</a>	1679
<a href="#">gdcmWin32.h</a>	1679
<a href="#">gdcmWLMFindQuery.h</a>	1680
<a href="#">gdcmWriter.h</a>	1681
<a href="#">gdcmXMLDictReader.h</a>	1682
<a href="#">gdcmXMLPrinter.h</a>	1682
<a href="#">gdcmXMLPrivateDictReader.h</a>	1683
<a href="#">vtkGDCMImageReader.h</a>	1684
<a href="#">vtkGDCMImageReader2.h</a>	1686
<a href="#">vtkGDCMImageWriter.h</a>	1687
<a href="#">vtkGDCMMedicalImageProperties.h</a>	1688
<a href="#">vtkGDCMPolyDataReader.h</a>	1688



<a href="#">vtkGDCMPolyDataWriter.h</a>	1689
<a href="#">vtkGDCMTesting.h</a>	1690
<a href="#">vtkGDCMThreadedImageReader.h</a>	1690
<a href="#">vtkGDCMThreadedImageReader2.h</a>	1691
<a href="#">vtkImageColorViewer.h</a>	1691
<a href="#">vtkImageMapToColors16.h</a>	1692
<a href="#">vtkImageMapToWindowLevelColors2.h</a>	1692
<a href="#">vtkImagePlanarComponentsToComponents.h</a>	1693
<a href="#">vtkImageRGBToYBR.h</a>	1693
<a href="#">vtkImageYBRToRGB.h</a>	1694
<a href="#">vtkLookupTable16.h</a>	1694
<a href="#">vtkRTStructSetProperties.h</a>	1695



## Chapter 9

# Namespace Documentation

### 9.1 gdcm Namespace Reference

#### Namespaces

- [network](#)
- [SegmentHelper](#)
- [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 [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 [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\_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 [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 [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 [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 [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<'\', 64, 0 >](#) [UIComp](#)
- typedef [String<'\', 64 >](#) [UTComp](#)

## Enumerations

- enum [CompOperators](#) {  
GDCM\_EQUAL = 0,  
GDCM\_DIFFERENT,  
GDCM\_GREATER,  
GDCM\_GREATEROREQUAL,  
GDCM\_LESS,  
GDCM\_LESSEOREQUAL }
- 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

- [ignore\\_char](#) const [backslash](#) ("\\")
- template<typename T >  
static int [Clamp](#) (T v)

- [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 [Version](#) &v)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [NestedModuleEntries](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [SwapCode](#) &sc)
- std::ostream & [operator<<](#) (std::ostream &os, const [FileSet](#) &f)
- std::ostream & [operator<<](#) (std::ostream &os, const [Region](#) &r)
- std::ostream & [operator<<](#) (std::ostream &os, [Event](#) &e)

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

- std::ostream & [operator<<](#) (std::ostream &os, const [PDBelement](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CommandDataSet](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [Orientation](#) &o)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODs](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Macros](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Modules](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Type](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [ModuleEntry](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [GroupDict](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [PrivateTag](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IOD](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [File](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Usage](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [MrProtocol](#) &d)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeaderDictEntry](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [Sorter](#) &s)
- std::ostream & [operator<<](#) (std::ostream &os, const [Preamble](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODEntry](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Macro](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeaderDict](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [Dicts](#) &d)
- std::ostream & [operator<<](#) (std::ostream &os, const [PDBHeader](#) &d)
- std::ostream & [operator<<](#) (std::ostream &os, const [CodeString](#) &str)
- std::ostream & [operator<<](#) (std::ostream &os, const [Directory](#) &d)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Module](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [Global](#) &g)
- std::ostream & [operator<<](#) (std::ostream &os, const [Object](#) &obj)
- std::ostream & [operator<<](#) (std::ostream &os, const [BasicOffsetTable](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [DictEntry](#) &val)
- std::ostream & [operator<<](#) (std::ostream &os, const [VL](#) &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 [TransferSyntax](#) &ts)
- std::ostream & [operator<<](#) (std::ostream &os, const [FileMetaInformation](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [VM](#) &\_val)
- std::ostream & [operator<<](#) (std::ostream &os, const [Scanner](#) &s)
- std::ostream & [operator<<](#) (std::ostream &os, const [StrictScanner](#) &s)
- std::ostream & [operator<<](#) (std::ostream &os, const [Dict](#) &val)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [MediaStorage](#) &ms)



- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `std::ostream & operator<< (std::ostream &_os, const VR &val)`
- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`
- `std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &os, const DataElement &val)`
- `std::ostream & operator<< (std::ostream &_os, const Tag &_val)`
- `std::ostream & operator<< (std::ostream &os, const Item &val)`
- `std::ostream & operator<< (std::ostream &os, const DataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateDict &val)`
- `std::ostream & operator<< (std::ostream &_os, const UIDs &uid)`
- `bool operator== (const CodeString &ref, const CodeString &cs)`
- `template<char TDelimiter, unsigned int TMaxLength, char TPadChar>  
std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)`
- `std::istream & operator>> (std::istream &in, ignore\_char const &ic)`
- `std::istream & operator>> (std::istream &_is, Tag &_val)`
- `template<typename T >  
static int Round (T x)`
- `template<typename Float >  
std::string to\_string (Float data)`
- `TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN`

## Variables

- static [Global GlobalInstance](#)
- [VRBINARY](#)

### 9.1.1 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.

## 9.1.2 Typedef Documentation

### 9.1.2.1 AECComp

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

### 9.1.2.2 ASComp

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

### 9.1.2.3 BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER

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

### 9.1.2.4 CSComp

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

### 9.1.2.5 DComp

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

### 9.1.2.6 DTComp

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

#### 9.1.2.7 FileList

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

#### 9.1.2.8 IconImage

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

#### 9.1.2.9 LOComp

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

#### 9.1.2.10 LTComp

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

#### 9.1.2.11 MacroEntry

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

#### 9.1.2.12 NestedMacroEntries

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

#### 9.1.2.13 PNComp

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

#### 9.1.2.14 SHComp

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

#### 9.1.2.15 STComp

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

#### 9.1.2.16 TMComp

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

#### 9.1.2.17 UIComp

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

#### 9.1.2.18 UTComp

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

### 9.1.3 Enumeration Type Documentation

#### 9.1.3.1 CompOperators

```
enum gdcM::CompOperators
```

##### Enumerator

GDCM_EQUAL	
GDCM_DIFFERENT	
GDCM_GREATER	
GDCM_GREATEROREQUAL	
GDCM_LESS	
GDCM_LESOREQUAL	

### 9.1.3.2 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	

### 9.1.3.3 ENQueryType

enum [gdcm::ENQueryType](#)

#### Enumerator

eCreateMMPS	
eSetMMPS	

### 9.1.3.4 EQueryLevel

enum [gdcm::EQueryLevel](#)

**Enumerator**

ePatient	
eStudy	
eSeries	
eImage	

**9.1.3.5 EQueryType**

enum [gdcm:EQueryType](#)

**Enumerator**

eFind	
eMove	
eWLMFind	

**9.1.3.6 ERootType**

enum [gdcm:ERootType](#)

**Enumerator**

ePatientRootType	
eStudyRootType	

**9.1.3.7 LodModeType**

enum [gdcm:LodModeType](#)

**Enumerator**

LD_ALL	
LD_NOSEQ	
LD_NOSHADOW	
LD_NOSHADOWSEQ	

## 9.1.4 Function Documentation

### 9.1.4.1 backslash()

```
ignore_char const gdcm::backslash (
    '\\ ' )
```

Referenced by `gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength()`.

### 9.1.4.2 Clamp()

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

Referenced by `gdcm::ImageChangePhotometricInterpretation::RGB2YBR()`, and `gdcm::ImageChangePhotometricInterpretation::YBR2RGB()`.

### 9.1.4.3 GetVRFromTag()

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

### 9.1.4.4 operator!=() [1/2]

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

### 9.1.4.5 operator!=() [2/2]

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

**9.1.4.6 operator<<()** [1/56]

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

References gdcm::Version::Print().

**9.1.4.7 operator<<()** [2/56]

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

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::↔Name.

**9.1.4.8 operator<<()** [3/56]

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

References gdcm::SwapCode::GetSwapCodeString().

**9.1.4.9 operator<<()** [4/56]

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

**9.1.4.10 operator<<()** [5/56]

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

References gdcm::Region::Print().



#### 9.1.4.11 operator<<() [6/56]

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

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

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

#### 9.1.4.12 operator<<() [7/56]

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

References [gdcm::PDBElement::NameField](#), and [gdcm::PDBElement::ValueField](#).

#### 9.1.4.13 operator<<() [8/56]

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

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

#### 9.1.4.14 operator<<() [9/56]

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

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

#### 9.1.4.15 operator<<() [10/56]

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

**9.1.4.16 operator<<()** [11/56]

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

**9.1.4.17 operator<<()** [12/56]

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

**9.1.4.18 operator<<()** [13/56]

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

References `gdcm::Type::GetTypeString()`.

**9.1.4.19 operator<<()** [14/56]

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

References `gdcm::ModuleEntry::DataElementType`, `gdcm::ModuleEntry::DescriptionField`, and `gdcm::ModuleEntry::↔Name`.

**9.1.4.20 operator<<()** [15/56]

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

References `gdcm::GroupDict::GetAbbreviation()`, `gdcm::GroupDict::GetName()`, and `gdcm::GroupDict::Size()`.

**9.1.4.21 operator<<()** [16/56]

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

**9.1.4.22 operator<<()** [17/56]

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

**9.1.4.23 operator<<()** [18/56]

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

References `gdcm::File::GetHeader()`.

**9.1.4.24 operator<<()** [19/56]

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

References `gdcm::Usage::GetUsageString()`.

**9.1.4.25 operator<<()** [20/56]

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

References `gdcm::MrProtocol::Print()`.

**9.1.4.26 operator<<()** [21/56]

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

**9.1.4.27 operator<<()** [22/56]

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

References gdcm::Sorter::Print().

**9.1.4.28 operator<<()** [23/56]

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

**9.1.4.29 operator<<()** [24/56]

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

**9.1.4.30 operator<<()** [25/56]

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

**9.1.4.31 operator<<()** [26/56]

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

**9.1.4.32 operator<<()** [27/56]

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

**9.1.4.33 operator<<()** [28/56]

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

References gdcm::PDBHeader::Print().

**9.1.4.34 operator<<()** [29/56]

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

**9.1.4.35 operator<<()** [30/56]

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

References gdcm::Directory::Print().

Referenced by gdcm::CSAElement::CSAElement(), gdcm::CSAHeaderDict::CSAHeaderDict(), gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry(), gdcm::Dict::Dict(), gdcm::DictEntry::DictEntry(), gdcm::VL::GetLength(), gdcm::TransferSyntax::GetString(), gdcm::IOD::IOD(), gdcm::IODEntry::IODEntry(), gdcm::IODs::IODs(), gdcm::Macro::Macro(), gdcm::Macros::Macros(), gdcm::Module::Module(), gdcm::Modules::Modules(), gdcm::NestedModuleEntries::NestedModuleEntries(), gdcm::MediaStorage::operator MType(), gdcm::Type::operator TypeType(), gdcm::Usage::operator UsageType(), gdcm::VM::operator VMType(), gdcm::PDBelement::PDBelement(), gdcm::PhotometricInterpretation::PhotometricInterpretation(), gdcm::SwapCode::SwapCode(), gdcm::Tag::Tag(), gdcm::VR::Write(), gdcm::CommandDataSet::~~CommandDataSet(), gdcm::GroupDict::~~GroupDict(), and gdcm::ModuleEntry::~~ModuleEntry().

**9.1.4.36 operator<<()** [31/56]

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

**9.1.4.37 operator<<()** [32/56]

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

References gdcm::PhotometricInterpretation::GetPIString().

**9.1.4.38 operator<<()** [33/56]

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

**9.1.4.39 operator<<()** [34/56]

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

References gdcm::Object::Print().

**9.1.4.40 operator<<()** [35/56]

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

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::ValueField, and gdcm::DataElement::Value↔LengthField.

Referenced by gdcm::BasicOffsetTable::BasicOffsetTable().

**9.1.4.41 operator<<()** [36/56]

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

**9.1.4.42 operator<<()** [37/56]

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

**9.1.4.43 operator<<()** [38/56]

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

References gdcm::CSAElement::DataField, gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::CSAElement::KeyField, gdcm::CSAElement::NameField, gdcm::CSAElement::NoOfItemsField, gdcm::CSAElement::SyngoDTField, gdcm::CSAElement::ValueMultiplicityField, gdcm::VM::VM1, and gdcm::CSAElement::VRField.

**9.1.4.44 operator<<()** [39/56]

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

References gdcm::CSAHeader::Print().

**9.1.4.45 operator<<()** [40/56]

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

References gdcm::TransferSyntax::GetTSSString().

**9.1.4.46 operator<<()** [41/56]

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

References gdcm::FileMetaInformation::GetPreamble(), and gdcm::DataSet::Print().

**9.1.4.47 operator<<()** [42/56]

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

References gdcm::VM::GetVMString().

**9.1.4.48 operator<<()** [43/56]

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

References gdcm::Scanner::Print().

**9.1.4.49 operator<<()** [44/56]

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

References gdcm::StrictScanner::Print().

**9.1.4.50 operator<<()** [45/56]

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



#### 9.1.4.51 operator<<() [46/56]

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

References [gdcm::MediaStorage::GetMSString\(\)](#).

#### 9.1.4.52 operator<<() [47/56]

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

References [gdcm::DataElement::TagField](#), [gdcm::DataElement::ValueField](#), and [gdcm::DataElement::ValueLengthField](#).

Referenced by [gdcm::Fragment::Fragment\(\)](#).

#### 9.1.4.53 operator<<() [48/56]

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

References [gdcm::VR::GetVRString\(\)](#).

#### 9.1.4.54 operator<<() [49/56]

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

References [gdcm::UI::Internal](#).

**9.1.4.55 operator<<()** [50/56]

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

References gdcmm::PixelFormat::Print().

**9.1.4.56 operator<<()** [51/56]

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

References gdcmm::Object::Print(), gdcmm::DataElement::TagField, gdcmm::DataElement::ValueField, gdcmm::DataElement::ValueLengthField, and gdcmm::DataElement::VRField.

Referenced by gdcmm::DataElement::DataElement().

**9.1.4.57 operator<<()** [52/56]

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

**9.1.4.58 operator<<()** [53/56]

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

References gdcmm::DataSet::Print(), gdcmm::DataElement::TagField, and gdcmm::DataElement::ValueLengthField.

Referenced by gdcmm::Item::Item().

#### 9.1.4.59 operator<<() [54/56]

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

References gdcm::DataSet::Begin(), and gdcm::DataSet::Print().

Referenced by gdcm::DataSet::InsertDataElement().

#### 9.1.4.60 operator<<() [55/56]

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

#### 9.1.4.61 operator<<() [56/56]

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

References gdcm::UIDs::GetName(), and gdcm::UIDs::GetString().

#### 9.1.4.62 operator==( )

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

Examples:

[DumpPhilipsECHO.cxx](#).

Referenced by gdcm::Value::~~Value().

**9.1.4.63 operator>>()** [1/3]

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

Referenced by gdcmm::Tag::Tag().

**9.1.4.64 operator>>()** [2/3]

```
std::istream& gdcmm::operator>> (
    std::istream & in,
    ignore_char const & ic ) [inline]
```

References gdcmm::ignore\_char::m\_char.

**9.1.4.65 operator>>()** [3/3]

```
std::istream& gdcmm::operator>> (
    std::istream & _is,
    Tag & _val ) [inline]
```

References gdcmm::Tag::SetElement(), and gdcmm::Tag::SetGroup().

**9.1.4.66 Round()**

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

Referenced by gdcmm::ImageChangePhotometricInterpretation::RGB2YBR(), and gdcmm::ImageChangePhotometricInterpretation::YBR2RGB().

**9.1.4.67 to\_string()**

```
template<typename Float >
std::string gdcmm::to_string (
    Float data )
```

Referenced by gdcmm::EncodingImplementation< VR::VRASCII >::Write().

#### 9.1.4.68 TYPETOENCODING()

```
gdcmm::TYPETOENCODING (
    SQ ,
    VRBINARY ,
    unsigned char )
```

### 9.1.5 Variable Documentation

#### 9.1.5.1 GlobalInstance

```
Global gdcmm::GlobalInstance [static]
```

#### 9.1.5.2 VRBINARY

```
gdcmm::VRBINARY
```

Referenced by gdcmm::Element< TVR, VM::VM1\_n >::Set(), and gdcmm::Element< TVR, VM::VM1\_n >::SetNoSwap().

## 9.2 gdcmm::network Namespace Reference

### Classes

- class [AAbortPDU](#)  
[AAbortPDU.](#)
- 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_RQPDURceivedOpen,  
eARELEASE_RPPDURceived,  
eARELEASEResponse,  
eAABORTRequest,  
eAABORTPDURceivedOpen,  
eTransportConnectionClosed,  
eARTIMTimerExpired,  
eUnrecognizedPDURceived,  
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

### 9.2.1 Enumeration Type Documentation

#### 9.2.1.1 EEventID

```
enum gdcm::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	

## 9.2.1.2 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	

## Enumerator

eSta9ReleaseCollisionRqLocal	
eSta10ReleaseCollisionAc	
eSta11ReleaseCollisionRq	
eSta12ReleaseCollisionAcLocal	
eSta13AwaitingClose	

## 9.2.2 Function Documentation

## 9.2.2.1 GetStateIndex()

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

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

## 9.2.3 Variable Documentation

## 9.2.3.1 cMaxEventID

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

## 9.2.3.2 cMaxStateID

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

Referenced by gdcmm::network::TableRow::TableRow(), and gdcmm::network::TableRow::~~TableRow().

## 9.3 gdcmm::SegmentHelper Namespace Reference

### Classes

- struct [BasicCodedEntry](#)  
*This structure defines a basic coded entry with all of its attributes.*

## 9.4 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)

### 9.4.1 Detailed Description

Class for Terminal.

Allow one to print in color in a shell

- support VT100 compatible shell
- win32 console

## 9.4.2 Enumeration Type Documentation

### 9.4.2.1 Attribute

enum `gdcmm::terminal::Attribute`

#### Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

### 9.4.2.2 Color

enum `gdcmm::terminal::Color`

#### Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

### 9.4.2.3 Mode

enum `gdcmm::terminal::Mode`

#### Enumerator

CONSOLE	
VT100	

### 9.4.3 Function Documentation

#### 9.4.3.1 setattribute()

```
GDCM_EXPORT std::string gdc::terminal::setattribute (
    Attribute att )
```

#### 9.4.3.2 setbgcolor()

```
GDCM_EXPORT std::string gdc::terminal::setbgcolor (
    Color c )
```

#### 9.4.3.3 setfgcolor()

```
GDCM_EXPORT std::string gdc::terminal::setfgcolor (
    Color c )
```

#### 9.4.3.4 setmode()

```
GDCM_EXPORT void gdc::terminal::setmode (
    Mode m )
```

## Chapter 10

# Class Documentation

### 10.1 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
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetReason](#) (const uint8\_t r)
- void [SetSource](#) (const uint8\_t s)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.1.1 Detailed Description

[AAbortPDU](#).

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

### 10.1.2 Constructor & Destructor Documentation

#### 10.1.2.1 AAbortPDU()

```
gdcmm::network::AAbortPDU::AAbortPDU ( )
```

### 10.1.3 Member Function Documentation

#### 10.1.3.1 IsLastFragment()

```
bool gdcmm::network::AAbortPDU::IsLastFragment ( ) const [inline], [virtual]
```

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

#### 10.1.3.2 Print()

```
void gdcmm::network::AAbortPDU::Print (
    std::ostream & os ) const [virtual]
```

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



### 10.1.3.3 Read()

```
std::istream& gdcm::network::AAabortPDU::Read (
    std::istream & is ) [virtual]
```

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

### 10.1.3.4 SetReason()

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

### 10.1.3.5 SetSource()

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

### 10.1.3.6 Size()

```
size_t gdcm::network::AAabortPDU::Size ( ) const [virtual]
```

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

### 10.1.3.7 Write()

```
const std::ostream& gdcm::network::AAabortPDU::Write (
    std::ostream & os ) const [virtual]
```

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

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

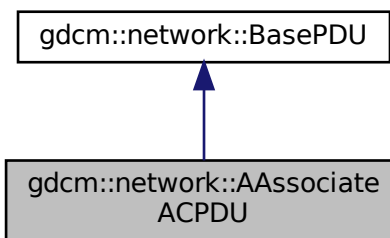
- [gdcmAAabortPDU.h](#)

## 10.2 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
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- [SizeType](#) [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

## Protected Member Functions

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

## Friends

- class [AAssociateRQPDU](#)

### 10.2.1 Detailed Description

[AAssociateACPDU](#).

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

### 10.2.2 Member Typedef Documentation

#### 10.2.2.1 SizeType

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

### 10.2.3 Constructor & Destructor Documentation

### 10.2.3.1 AAssociateACPDU()

```
gdcM::network::AAssociateACPDU::AAssociateACPDU ( )
```

## 10.2.4 Member Function Documentation

### 10.2.4.1 AddPresentationContextAC()

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

### 10.2.4.2 GetNumberOfPresentationContextAC()

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

### 10.2.4.3 GetPresentationContextAC()

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

### 10.2.4.4 GetUserInfoInformation()

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

References Print(), and Size().

### 10.2.4.5 InitFromRQ()

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

Referenced by IsLastFragment().

#### 10.2.4.6 IsLastFragment()

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

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

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

#### 10.2.4.7 Print()

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

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

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

#### 10.2.4.8 Read()

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

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

#### 10.2.4.9 SetCalledAETitle()

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

#### 10.2.4.10 SetCallingAETitle()

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

#### 10.2.4.11 Size()

```
SizeType gdcmm::network::AAssociateACPDU::Size ( ) const [virtual]
```

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

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

#### 10.2.4.12 Write()

```
const std::ostream& gdcmm::network::AAssociateACPDU::Write (
    std::ostream & os ) const [virtual]
```

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

### 10.2.5 Friends And Related Function Documentation

#### 10.2.5.1 AAssociateRQPDU

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

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

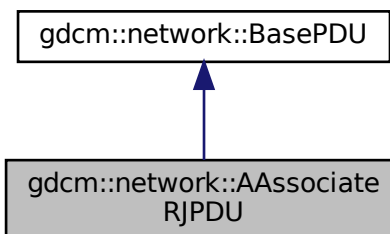
- [gdcmmAAssociateACPDU.h](#)

## 10.3 gdcmm::network::AAssociateRJPDU Class Reference

[AAssociateRJPDU](#).

```
#include <gdcmmAAssociateRJPDU.h>
```

Inheritance diagram for [gdcmm::network::AAssociateRJPDU](#):



Collaboration diagram for gdcm::network::AAssociateRJPDU:



### Public Member Functions

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

#### 10.3.1 Detailed Description

[AAssociateRJPDU](#).

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

#### 10.3.2 Constructor & Destructor Documentation

##### 10.3.2.1 AAssociateRJPDU()

```
gdcm::network::AAssociateRJPDU::AAssociateRJPDU ( )
```

#### 10.3.3 Member Function Documentation

#### 10.3.3.1 IsLastFragment()

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

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

#### 10.3.3.2 Print()

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

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

#### 10.3.3.3 Read()

```
std::istream& gdcn::network::AAssociateRJPDU::Read (
    std::istream & is ) [virtual]
```

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

#### 10.3.3.4 Size()

```
size_t gdcn::network::AAssociateRJPDU::Size ( ) const [virtual]
```

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

#### 10.3.3.5 Write()

```
const std::ostream& gdcn::network::AAssociateRJPDU::Write (
    std::ostream & os ) const [virtual]
```

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

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

- [gdcnAAssociateRJPDU.h](#)



## 10.4 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

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

## 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 [AAssociateACPDU](#)

### 10.4.1 Detailed Description

[AAssociateRQPDU](#).

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

### 10.4.2 Member Typedef Documentation

#### 10.4.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContext↵
ArrayType
```

#### 10.4.2.2 SizeType

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

### 10.4.3 Constructor & Destructor Documentation

#### 10.4.3.1 AAssociateRQPDU() [1/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )
```

#### 10.4.3.2 AAssociateRQPDU() [2/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU (
    const AAssociateRQPDU & pdu ) [inline]
```

### 10.4.4 Member Function Documentation

#### 10.4.4.1 AddPresentationContext()

```
void gdcm::network::AAssociateRQPDU::AddPresentationContext (
    PresentationContextRQ const & pc )
```

#### 10.4.4.2 GetCalledAETitle()

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

References SetCallingAETitle().

#### 10.4.4.3 GetCallingAETitle()

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

References IsAETitleValid(), and Print().

#### 10.4.4.4 GetNumberOfPresentationContext()

```
SizeType gdcn::network::AAssociateRQPDU::GetNumberOfPresentationContext ( ) const [inline]
```

#### 10.4.4.5 GetPresentationContext()

```
PresentationContextRQ const& gdcn::network::AAssociateRQPDU::GetPresentationContext (
    SizeType i ) const [inline]
```

#### 10.4.4.6 GetPresentationContextByAbstractSyntax()

```
const PresentationContextRQ* gdcn::network::AAssociateRQPDU::GetPresentationContextByAbstract←
Syntax (
    AbstractSyntax const & absyn ) const
```

Referenced by GetPresentationContexts().

#### 10.4.4.7 GetPresentationContextByID()

```
const PresentationContextRQ* gdcn::network::AAssociateRQPDU::GetPresentationContextByID (
    uint8_t i ) const
```

Referenced by GetPresentationContexts().

#### 10.4.4.8 GetPresentationContexts()

```
PresentationContextArrayType const& gdcn::network::AAssociateRQPDU::GetPresentationContexts ( )
[inline]
```

References GetPresentationContextByAbstractSyntax(), and GetPresentationContextByID().

#### 10.4.4.9 GetReserved43\_74()

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

#### 10.4.4.10 GetUserInfoInformation()

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

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

#### 10.4.4.11 IsAETitleValid()

```
static bool gdcm::network::AAssociateRQPDU::IsAETitleValid (
    const char title[16] ) [static]
```

Check whether or not the.

##### Parameters

<i>title</i>	is a valid AE title
--------------	---------------------

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

#### 10.4.4.12 IsLastFragment()

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

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

#### 10.4.4.13 Print()

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

This function will initialize an [AAssociateACPDU](#) from the fields in the [AAssociateRQPDU](#) structure

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

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

#### 10.4.4.14 Read()

```
std::istream& gdcm::network::AAssociateRQPDU::Read (
    std::istream & is ) [virtual]
```

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

#### 10.4.4.15 SetCalledAETitle()

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

Set the Called AE Title.

#### 10.4.4.16 SetCallingAETitle()

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

Set the Calling AE Title.

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

#### 10.4.4.17 SetUserInfoInformation()

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

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

#### 10.4.4.18 Size()

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

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

## 10.4.4.19 Write()

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

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

## 10.4.5 Friends And Related Function Documentation

## 10.4.5.1 AAssociateACPDU

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

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

- [gdcmAAssociateRQPDU.h](#)

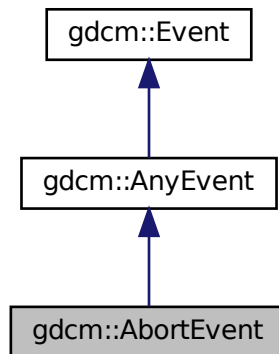
## 10.5 gdcm::AbortEvent Class Reference

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

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for `gdcm::AbortEvent`:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.6 `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



## 10.6.1 Detailed Description

[AbstractSyntax](#).

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

## 10.6.2 Constructor & Destructor Documentation

### 10.6.2.1 AbstractSyntax()

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

## 10.6.3 Member Function Documentation

### 10.6.3.1 GetAsDataElement()

```
DataElement gdcm::network::AbstractSyntax::GetAsDataElement ( ) const
```

Referenced by `operator==( )`.

### 10.6.3.2 GetName()

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

References `Print()`, `SetNameFromUID()`, and `Size()`.

### 10.6.3.3 operator==( )

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

References `GetAsDataElement()`.

#### 10.6.3.4 Print()

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

Referenced by GetName().

#### 10.6.3.5 Read()

```
std::istream& gdcn::network::AbstractSyntax::Read (
    std::istream & is )
```

#### 10.6.3.6 SetName()

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

#### 10.6.3.7 SetNameFromUID()

```
void gdcn::network::AbstractSyntax::SetNameFromUID (
    UIDs::TSName tsname )
```

Referenced by GetName().

#### 10.6.3.8 Size()

```
size_t gdcn::network::AbstractSyntax::Size ( ) const
```

Referenced by GetName().

#### 10.6.3.9 Write()

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

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

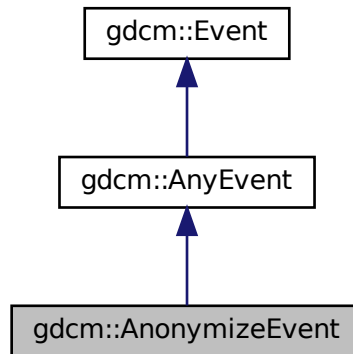
- [gdcnAbstractSyntax.h](#)

## 10.7 gdcm::AnonymizeEvent Class Reference

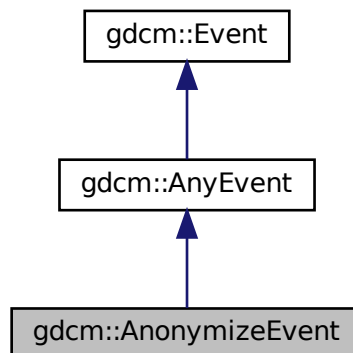
[AnonymizeEvent](#).

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

### 10.7.1 Detailed Description

[AnonymizeEvent](#).

Special type of event triggered during the Anonymization process

See also

[Anonymizer](#)

### 10.7.2 Member Typedef Documentation

#### 10.7.2.1 Self

```
typedef AnonymizeEvent gdcm::AnonymizeEvent::Self
```

#### 10.7.2.2 Superclass

```
typedef AnyEvent gdcm::AnonymizeEvent::Superclass
```

### 10.7.3 Constructor & Destructor Documentation

#### 10.7.3.1 AnonymizeEvent() [1/2]

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

### 10.7.3.2 ~AnonymizeEvent()

```
virtual gdcm::AnonymizeEvent::~~AnonymizeEvent ( ) [inline], [virtual]
```

### 10.7.3.3 AnonymizeEvent() [2/2]

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

## 10.7.4 Member Function Documentation

### 10.7.4.1 CheckEvent()

```
virtual bool gdcm::AnonymizeEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [virtual]
```

### 10.7.4.2 GetEventName()

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

### 10.7.4.3 GetTag()

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

### 10.7.4.4 MakeObject()

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

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

Implements [gdcm::Event](#).

#### 10.7.4.5 SetTag()

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

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

- [gdcMAnonymizeEvent.h](#)

## 10.8 gdcM::Anonymizer Class Reference

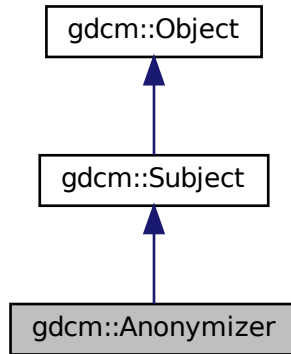
[Anonymizer.](#)

```
#include <gdcMAnonymizer.h>
```

Inheritance diagram for gdcM::Anonymizer:



Collaboration diagram for gdcmm::Anonymizer:



## Public Member Functions

- [Anonymizer](#) ()
- [~Anonymizer](#) ()
- bool [BasicApplicationLevelConfidentialityProfile](#) (bool deidentify=true)
- bool [Empty](#) (Tag const &t)
- const [CryptographicMessageSyntax](#) \* [GetCryptographicMessageSyntax](#) () const
- [File](#) & [GetFile](#) ()
- bool [Remove](#) (Tag const &t)
- 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](#) (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.*

## 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)

### 10.8.1 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  $O(m*\log(n))$  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](#), and [EncapsulateFileInRawData.cxx](#).

## 10.8.2 Constructor & Destructor Documentation

### 10.8.2.1 Anonymizer()

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

### 10.8.2.2 ~Anonymizer()

```
gdcm::Anonymizer::~~Anonymizer ( )
```

## 10.8.3 Member Function Documentation

### 10.8.3.1 BALCPPProtect()

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

### 10.8.3.2 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

### 10.8.3.3 CanEmptyTag()

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

### 10.8.3.4 ClearInternalUIDs()

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

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

#### Warning

the mapping is definitely lost

### 10.8.3.5 Empty()

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

Make [Tag](#) t empty (if not found tag will be created) Warning: does not handle SQ element

#### Examples:

[CreateJPIPDataSet.cxx](#).

### 10.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes()

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

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

Examples:

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

### 10.8.3.7 GetCryptographicMessageSyntax()

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

### 10.8.3.8 GetFile()

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

### 10.8.3.9 New()

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

for wrapped language: instantiate a reference counted object

### 10.8.3.10 RecurseDataSet()

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

### 10.8.3.11 Remove()

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

remove a tag (even a SQ can be removed) Return code is false when tag t cannot be found

#### 10.8.3.12 RemoveGroupLength()

```
bool gdcm::Anonymizer::RemoveGroupLength ( )
```

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

#### 10.8.3.13 RemovePrivateTags()

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

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

#### 10.8.3.14 RemoveRetired()

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

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

#### 10.8.3.15 Replace() [1/2]

```
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](#), and [EncapsulateFileInRawData.cxx](#).

### 10.8.3.16 Replace() [2/2]

```
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

### 10.8.3.17 SetCryptographicMessageSyntax()

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

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

### 10.8.3.18 SetFile()

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

Set/Get [File](#).

#### Examples:

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

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

- [gdcmAnonymizer.h](#)

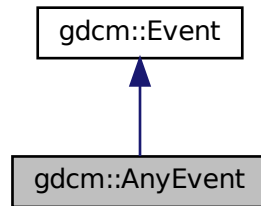
## 10.9 gdcmm::AnyEvent Class Reference

```
#include <gdcmmEvent.h>
```

Inheritance diagram for gdcmm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.10 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

### 10.10.1 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 )

## 10.10.2 Constructor & Destructor Documentation

### 10.10.2.1 ApplicationContext()

```
gdcm::network::ApplicationContext::ApplicationContext ( )
```

## 10.10.3 Member Function Documentation

### 10.10.3.1 GetName()

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

References Print(), and Size().

### 10.10.3.2 Print()

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

Referenced by GetName().

### 10.10.3.3 Read()

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

### 10.10.3.4 SetName()

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



## 10.10.3.5 Size()

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

Referenced by GetName().

## 10.10.3.6 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](#)

## 10.11 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](#) = ''

### 10.11.1 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

### 10.11.2 Member Function Documentation

#### 10.11.2.1 IsValid()

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

#### 10.11.2.2 Print()

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

#### 10.11.2.3 SetBlob()

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

#### 10.11.2.4 Squeeze()

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

### 10.11.3 Member Data Documentation

#### 10.11.3.1 Internal

```
std::string gdcm::ApplicationEntity::Internal
```

#### 10.11.3.2 MaxLength

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

#### 10.11.3.3 MaxNumberOfComponents

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

#### 10.11.3.4 Padding

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

#### 10.11.3.5 Separator

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

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

- [gdcMApplicationEntity.h](#)

## 10.12 gdcM::network::AReleaseRPPDU Class Reference

[AReleaseRPPDU](#).

```
#include <gdcMAReleaseRPPDU.h>
```

Inheritance diagram for gdcM::network::AReleaseRPPDU:



Collaboration diagram for gdcM::network::AReleaseRPPDU:



## Public Member Functions

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

### 10.12.1 Detailed Description

[AReleaseRPPDU](#).

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

### 10.12.2 Constructor & Destructor Documentation

#### 10.12.2.1 AReleaseRPPDU()

```
gdcmm::network::AReleaseRPPDU::AReleaseRPPDU ( )
```

### 10.12.3 Member Function Documentation

#### 10.12.3.1 IsLastFragment()

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

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

#### 10.12.3.2 Print()

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

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

### 10.12.3.3 Read()

```
std::istream& gdcn::network::AReleaseRPPDU::Read (  
    std::istream & is ) [virtual]
```

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

### 10.12.3.4 Size()

```
size_t gdcn::network::AReleaseRPPDU::Size ( ) const [virtual]
```

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

### 10.12.3.5 Write()

```
const std::ostream& gdcn::network::AReleaseRPPDU::Write (  
    std::ostream & os ) const [virtual]
```

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

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

- [gdcnAReleaseRPPDU.h](#)

## 10.13 gdcn::network::AReleaseRQPDU Class Reference

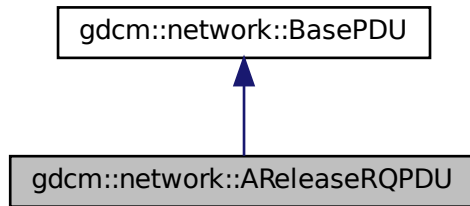
[AReleaseRQPDU](#).

```
#include <gdcnAReleaseRQPDU.h>
```

Inheritance diagram for gdcn::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



### Public Member Functions

- [AReleaseRQPDU](#) ()
- bool [IsLastFragment](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

#### 10.13.1 Detailed Description

[AReleaseRQPDU](#).

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

#### 10.13.2 Constructor & Destructor Documentation

##### 10.13.2.1 AReleaseRQPDU()

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

#### 10.13.3 Member Function Documentation

#### 10.13.3.1 IsLastFragment()

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

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

#### 10.13.3.2 Print()

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

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

#### 10.13.3.3 Read()

```
std::istream& gdcn::network::AReleaseRQPDU::Read (
    std::istream & is ) [virtual]
```

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

#### 10.13.3.4 Size()

```
size_t gdcn::network::AReleaseRQPDU::Size ( ) const [virtual]
```

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

#### 10.13.3.5 Write()

```
const std::ostream& gdcn::network::AReleaseRQPDU::Write (
    std::ostream & os ) const [virtual]
```

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

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

- [gdcnAReleaseRQPDU.h](#)



## 10.14 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](#) ()

### 10.14.1 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.

### 10.14.2 Constructor & Destructor Documentation

#### 10.14.2.1 ARTIMTimer()

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

### 10.14.3 Member Function Documentation

#### 10.14.3.1 GetElapsedTime()

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

#### 10.14.3.2 GetHasExpired()

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

#### 10.14.3.3 GetTimeout()

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

#### 10.14.3.4 SetTimeout()

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

#### 10.14.3.5 Start()

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

#### 10.14.3.6 Stop()

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

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

- [gdcMARTIMTimer.h](#)

## 10.15 gdcM::ASN1 Class Reference

Class for [ASN1](#).

```
#include <gdcMASN1.h>
```

## Public Member Functions

- [ASN1](#) ()
- [~ASN1](#) ()

## Static Public Member Functions

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

## Protected Member Functions

- int [TestPBKDF2](#) ()

### 10.15.1 Detailed Description

Class for [ASN1](#).

### 10.15.2 Constructor & Destructor Documentation

#### 10.15.2.1 ASN1()

```
gdcmm::ASN1::ASN1 ( )
```

#### 10.15.2.2 ~ASN1()

```
gdcmm::ASN1::~~ASN1 ( )
```

### 10.15.3 Member Function Documentation

#### 10.15.3.1 ParseDump()

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

### 10.15.3.2 ParseDumpFile()

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

### 10.15.3.3 TestPBKDF2()

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

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

- [gdcMASN1.h](#)

## 10.16 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

### 10.16.1 Detailed Description

[AsynchronousOperationsWindowSub.](#)

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

### 10.16.2 Constructor & Destructor Documentation

#### 10.16.2.1 AsynchronousOperationsWindowSub()

```
gdcM::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )
```

### 10.16.3 Member Function Documentation

#### 10.16.3.1 Print()

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

#### 10.16.3.2 Read()

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

#### 10.16.3.3 Size()

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

#### 10.16.3.4 Write()

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

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

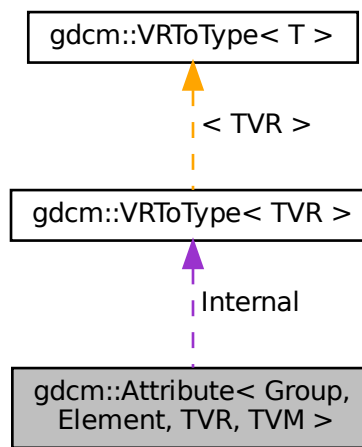
- [gdcmAsynchronousOperationsWindowSub.h](#)

## 10.17 gdcm::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 <gdcmAttribute.h>
```

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::VRType`)(`TagToType< Group, Element >::VRType`)))
- `GDCM_STATIC_ASSERT` (((`VM::VMType`) TVM &(`VM::VMType`)(`TagToType< Group, Element >::VMType`)))
- `GDCM_STATIC_ASSERT` (((((`VR::VRType`) TVR &`VR::VR_VM1`) &&((`VM::VMType`) TVM==`VM::VM1`))||!((`VR←::VRType`) TVR &`VR::VR_VM1`)))
- `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)

## 10.17.1 Detailed Description

```
template<uint16_t Group, uint16_t Element, int 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::VM2> 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`, `gdcmrtonplan.cxx`, `gdcmrtpplan.cxx`, `GenFake_← IdentifyFile.cxx`, `GetSequenceUltrasound.cxx`, `HelloWorld.cxx`, `LargeVRDSExplicit.cxx`, `PatchFile.cxx`, `pmsct_← rgb1.cxx`, `ReadAndDumpDICOMDIR2.cxx`, `ReadAndPrintAttributes.cxx`, `rle2img.cxx`, `SortImage.cxx`, `Stream_← ImageReaderTest.cxx`, and `VolumeSorter.cxx`.

## 10.17.2 Member Typedef Documentation

### 10.17.2.1 ArrayType

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

## 10.17.3 Member Enumeration Documentation

### 10.17.3.1 anonymous enum

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

#### Enumerator

VMType	
--------	--

## 10.17.4 Member Function Documentation

### 10.17.4.1 GDCM\_STATIC\_ASSERT() [1/3]

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

### 10.17.4.2 GDCM\_STATIC\_ASSERT() [2/3]

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



## 10.17.4.3 GDCM\_STATIC\_ASSERT() [3/3]

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

## 10.17.4.4 GetAsDataElement()

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

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

## 10.17.4.5 GetDictVM()

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

## 10.17.4.6 GetDictVR()

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

## 10.17.4.7 GetNumberOfValues()

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

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<().

**10.17.4.8 GetTag()**

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
static Tag gdcM::Attribute< Group, Element, TVR, TVM >::GetTag ( ) [inline], [static]
```

**10.17.4.9 GetValue()** [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
ArrayType& gdcM::Attribute< Group, Element, TVR, TVM >::GetValue (
    unsigned int idx = 0 ) [inline]
```

**10.17.4.10 GetValue()** [2/2]

```
template<uint16_t Group, uint16_t Element, int 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]
```

**10.17.4.11 GetValues()**

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
const ArrayType* gdcM::Attribute< Group, Element, TVR, TVM >::GetValues ( ) const [inline]
```

Referenced by gdcM::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcM::Attribute< Group, Element, T↵VR, VM::VM1 >::operator!=(), gdcM::Attribute< Group, Element, TVR, TVM >::operator<(), gdcM::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcM::Attribute< Group, Element, TVR, TVM >::operator==(), and gdcM::↵Attribute< Group, Element, TVR, VM::VM1 >::operator==().

**10.17.4.12 GetVM()**

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

## 10.17.4.13 GetVR()

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

## 10.17.4.14 operator!=(())

```
template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

## 10.17.4.15 operator&lt;()

```
template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

## 10.17.4.16 operator==(())

```
template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

## 10.17.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
ArrayType& gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx ) [inline]
```

**10.17.4.18 operator[]()** [2/2]

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
ArrayType const& gdcmm::Attribute< Group, Element, TVR, TVM >::operator[] (
    unsigned int idx ) const [inline]
```

**10.17.4.19 Print()**

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::Print (
    std::ostream & os ) const [inline]
```

**10.17.4.20 Set()**

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::Set (
    DataSet const & ds ) [inline]
```

References gdcmm::DataSet::GetDataElement().

**10.17.4.21 SetByteValue()**

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
void gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References gdcmm::ByteValue::GetLength(), and gdcmm::ByteValue::GetPointer().

**10.17.4.22 SetByteValueNoSwap()**

```
template<uint16_t Group, uint16_t Element, int 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::ByteValue::GetLength(), and gdcmm::ByteValue::GetPointer().

## 10.17.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
    DataElement const & de ) [inline]
```

Examples:

[ReadAndDumpDICOMDIR2.cxx](#).

References [gdcm::DataElement::GetByteValue\(\)](#), [gdcm::Tag::GetGroup\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVR\(\)](#), and [gdcm::DataElement::IsEmpty\(\)](#).

## 10.17.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

References [gdcm::DataSet::FindDataElement\(\)](#), [gdcm::DataSet::GetDataElement\(\)](#), and [gdcm::DataElement::IsEmpty\(\)](#).

## 10.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, int 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]
```

## 10.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, int 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:

[LargeVRDSExplicit.cxx](#).

## 10.17.5 Member Data Documentation

### 10.17.5.1 Internal

```
template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM =
TagToType<Group, Element>::VMType>
ArrayType gdcM::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by `gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM()`.

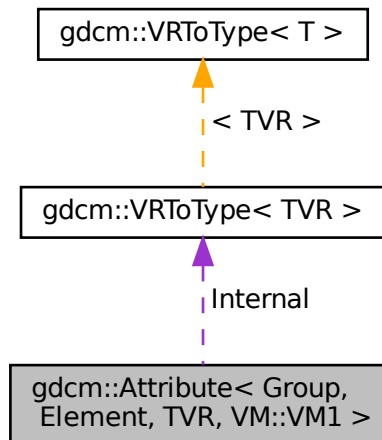
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

## 10.18 `gdcM::Attribute< Group, Element, TVR, VM::VM1 >` Class Template Reference

```
#include <gdcMAttribute.h>
```

Collaboration diagram for `gdcM::Attribute< Group, Element, TVR, VM::VM1 >`:



### Public Types

- enum { `VMType` = `VMToLength<VM::VM1>::Length` }
- typedef `VRToType< TVR >::Type` `ArrayType`

## Public Member Functions

- `GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)`
- `GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))`
- `GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))`
- `GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))`
- `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`
- `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)`

## 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)`

### 10.18.1 Member Typedef Documentation

#### 10.18.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, int TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType
```

## 10.18.2 Member Enumeration Documentation

### 10.18.2.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, int TVR>
anonymous enum
```

#### Enumerator

VMType	
--------	--

## 10.18.3 Member Function Documentation

### 10.18.3.1 GDCM\_STATIC\_ASSERT() [1/4]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    VMToLength< VM::VM1 >::Length  = =1 )
```

### 10.18.3.2 GDCM\_STATIC\_ASSERT() [2/4]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR & (VR::VRType) (TagToType< Group, Element >::VRType)) )
```

### 10.18.3.3 GDCM\_STATIC\_ASSERT() [3/4]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
    ((VM::VMType) VM::VM1 & (VM::VMType) (TagToType< Group, Element >::VMType)) )
```



## 10.18.3.4 GDCM\_STATIC\_ASSERT() [4/4]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::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)) )
```

## 10.18.3.5 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, int TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement ( ) const [inline]
```

References gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), and gdcm::DataElement::SetVR().

## 10.18.3.6 GetDictVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM ( ) [inline], [static]
```

## 10.18.3.7 GetDictVR()

```
template<uint16_t Group, uint16_t Element, int TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR ( ) [inline], [static]
```

## 10.18.3.8 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, int TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues ( ) const [inline]
```

## 10.18.3.9 GetTag()

```
template<uint16_t Group, uint16_t Element, int TVR>
static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag ( ) [inline], [static]
```

**10.18.3.10 GetValue()** [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) [inline]
```

**10.18.3.11 GetValue()** [2/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) const [inline]
```

**10.18.3.12 GetValues()**

```
template<uint16_t Group, uint16_t Element, int TVR>
const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues ( ) const [inline]
```

**10.18.3.13 GetVM()**

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM ( ) [inline], [static]
```

**10.18.3.14 GetVR()**

```
template<uint16_t Group, uint16_t Element, int TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR ( ) [inline], [static]
```

**10.18.3.15 operator!=(())**

```
template<uint16_t Group, uint16_t Element, int TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (
    const Attribute< Group, Element, TVR, VM::VM1 > & att ) const [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

**10.18.3.16 operator<()**

```
template<uint16_t Group, uint16_t Element, int 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(), and gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues().

**10.18.3.17 operator==(**

```
template<uint16_t Group, uint16_t Element, int 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 >::GetValues().

**10.18.3.18 Print()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print (
    std::ostream & os ) const [inline]
```

**10.18.3.19 Set()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set (
    DataSet const & ds ) [inline]
```

References gdcmm::DataSet::GetDataElement().

**10.18.3.20 SetByteValue()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References gdcmm::ByteValue::GetLength(), and gdcmm::ByteValue::GetPointer().

**10.18.3.21 SetByteValueNoSwap()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (
    const ByteValue * bv ) [inline], [protected]
```

References `gdcm::ByteValue::GetLength()`, and `gdcm::ByteValue::GetPointer()`.

**10.18.3.22 SetFromDataElement()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (
    DataElement const & de ) [inline]
```

References `gdcm::DataElement::GetByteValue()`, `gdcm::Tag::GetGroup()`, `gdcm::DataElement::GetTag()`, `gdcm::DataElement::GetVR()`, and `gdcm::DataElement::IsEmpty()`.

**10.18.3.23 SetFromDataSet()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

References `gdcm::DataSet::FindDataElement()`, `gdcm::DataSet::GetDataElement()`, and `gdcm::DataElement::IsEmpty()`.

**10.18.3.24 SetValue()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (
    ArrayType v ) [inline]
```

**10.18.4 Member Data Documentation**

## 10.18.4.1 Internal

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal
```

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

- [gdcmAttribute.h](#)

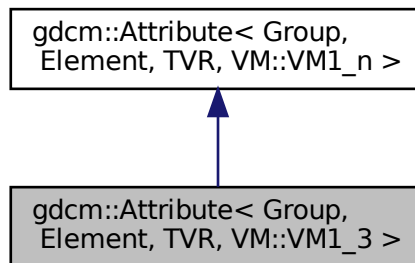
## 10.19 gdcm::Attribute&lt; Group, Element, TVR, VM::VM1\_3 &gt; Class Template Reference

```
#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 Member Functions

- [VM GetVM](#) () const

## Additional Inherited Members

### 10.19.1 Member Function Documentation

#### 10.19.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
VM gdcM::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const [inline]
```

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

- [gdcMAttribute.h](#)

## 10.20 gdcM::Attribute< Group, Element, TVR, VM::VM1\_8 > Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1\_8 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1\_8 >:



### Public Member Functions

- [VM GetVM](#) () const

### Additional Inherited Members

#### 10.20.1 Member Function Documentation

##### 10.20.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const [inline]
```

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

- [gdcmAttribute.h](#)

## 10.21 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 >:



### Public Types

- typedef `VRToType< TVR >::Type ArrayType`

### Public Member Functions

- `Attribute ()`
- `~Attribute ()`
- `GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))`
- `GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))`
- `GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))`
- `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`
- `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 (unsigned int idx, ArrayType v)`
- `void SetValue (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)

### 10.21.1 Member Typedef Documentation

#### 10.21.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, int TVR>
typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::ArrayType
```

### 10.21.2 Constructor & Destructor Documentation

#### 10.21.2.1 Attribute()

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Attribute ( ) [inline], [explicit]
```

#### 10.21.2.2 ~Attribute()

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::~Attribute ( ) [inline]
```

### 10.21.3 Member Function Documentation

**10.21.3.1 GDCM\_STATIC\_ASSERT()** [1/3]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    ((VR::VRType) TVR & (VR::VRType) (TagToType< Group, Element >::VRType)) )
```

**10.21.3.2 GDCM\_STATIC\_ASSERT()** [2/3]

```
template<uint16_t Group, uint16_t Element, int TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
    (VM::VM1_n & (VM::VMType) (TagToType< Group, Element >::VMType)) )
```

**10.21.3.3 GDCM\_STATIC\_ASSERT()** [3/3]

```
template<uint16_t Group, uint16_t Element, int 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::VRType) TVR & VR::VR_VM1)) )
```

**10.21.3.4 GetAsDataElement()**

```
template<uint16_t Group, uint16_t Element, int TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]
```

References `gdcm::DataElement::GetVR()`, `gdcm::DataElement::SetByteValue()`, and `gdcm::DataElement::SetVR()`.

**10.21.3.5 GetDictVM()**

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( ) [inline], [static]
```

**10.21.3.6 GetDictVR()**

```
template<uint16_t Group, uint16_t Element, int TVR>
static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( ) [inline], [static]
```

### 10.21.3.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, int TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const [inline]
```

### 10.21.3.8 GetTag()

```
template<uint16_t Group, uint16_t Element, int TVR>
static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( ) [inline], [static]
```

### 10.21.3.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.21.3.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

### 10.21.3.11 GetValues()

```
template<uint16_t Group, uint16_t Element, int TVR>
const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const [inline]
```

### 10.21.3.12 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( ) [inline], [static]
```

**10.21.3.13 GetVR()**

```
template<uint16_t Group, uint16_t Element, int TVR>
static VR gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( ) [inline], [static]
```

**10.21.3.14 operator[]()** [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType& gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) [inline]
```

**10.21.3.15 operator[]()** [2/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
ArrayType const& gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) const [inline]
```

**10.21.3.16 Print()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::Print (
    std::ostream & os ) const [inline]
```

**10.21.3.17 Set()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::Set (
    DataSet const & ds ) [inline]
```

References gdcM::DataSet::GetDataElement().

## 10.21.3.18 SetByteValue()

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (
    const ByteValue * bv ) [inline], [protected]
```

References `gdcm::ByteValue::GetLength()`, and `gdcm::ByteValue::GetPointer()`.

## 10.21.3.19 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (
    DataElement const & de ) [inline]
```

References `gdcm::DataElement::GetByteValue()`, `gdcm::Tag::GetGroup()`, `gdcm::DataElement::GetTag()`, `gdcm::DataElement::GetVR()`, and `gdcm::DataElement::IsEmpty()`.

## 10.21.3.20 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (
    DataSet const & ds ) [inline]
```

References `gdcm::DataSet::FindDataElement()`, `gdcm::DataSet::GetDataElement()`, and `gdcm::DataElement::IsEmpty()`.

## 10.21.3.21 SetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (
    unsigned int numel ) [inline]
```

## 10.21.3.22 SetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    unsigned int idx,
    ArrayType v ) [inline]
```

**10.21.3.23 SetValue()** [2/2]

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
    ArrayType v ) [inline]
```

References SetValue().

Referenced by SetValue().

**10.21.3.24 SetValues()**

```
template<uint16_t Group, uint16_t Element, int TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
    const ArrayType * array,
    unsigned int numel,
    bool own = false ) [inline]
```

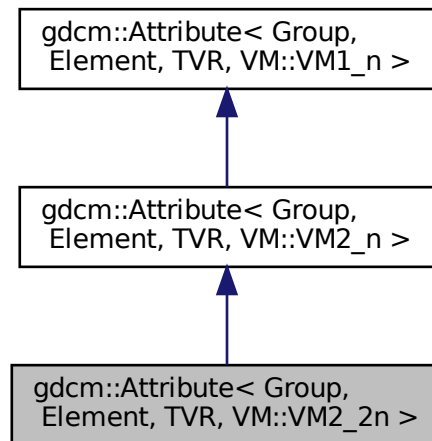
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

**10.22 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 gdcm::Attribute< Group, Element, TVR, VM::VM2\_2n >:



### Static Public Member Functions

- static [VM GetVM](#) ()

### Additional Inherited Members

#### 10.22.1 Member Function Documentation

##### 10.22.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( ) [inline], [static]
```

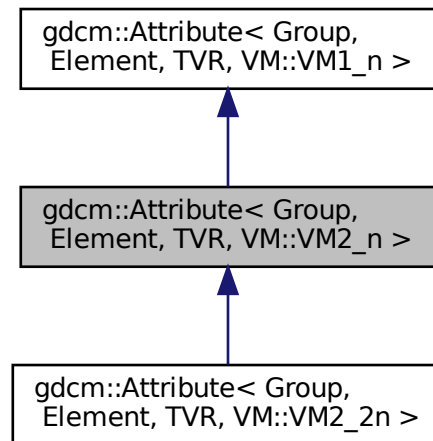
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

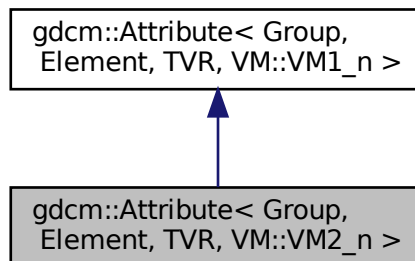
## 10.23 gdcm::Attribute< Group, Element, TVR, VM::VM2\_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2\_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2\_n >:



### Public Member Functions

- [VM GetVM](#) () const



## Additional Inherited Members

### 10.23.1 Member Function Documentation

#### 10.23.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const [inline]
```

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

- [gdcmAttribute.h](#)

## 10.24 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 >`:



## Static Public Member Functions

- static [VM GetVM](#) ()

## Additional Inherited Members

### 10.24.1 Member Function Documentation

#### 10.24.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>
static VM gdcm::Attribute< Group, Element, TVR, VM::VM3\_3n >::GetVM ( ) [inline], [static]
```

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

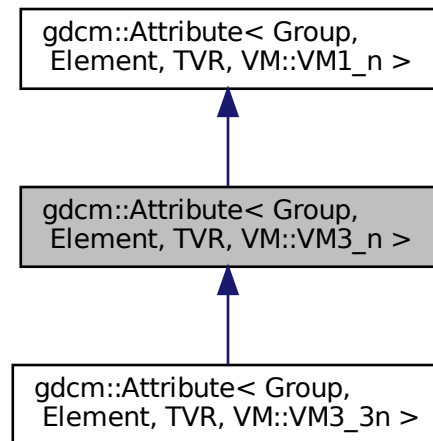
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 10.25 gdcm::Attribute< Group, Element, TVR, VM::VM3\_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3\_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3\_n >:



### Static Public Member Functions

- static [VM GetVM](#) ()

## Additional Inherited Members

### 10.25.1 Member Function Documentation

#### 10.25.1.1 GetVM()

```
template<uint16_t Group, uint16_t Element, int TVR>  
static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM ( ) [inline], [static]
```

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

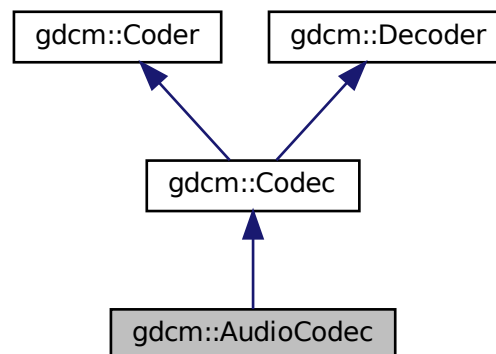
- [gdcMAttribute.h](#)

## 10.26 gdcM::AudioCodec Class Reference

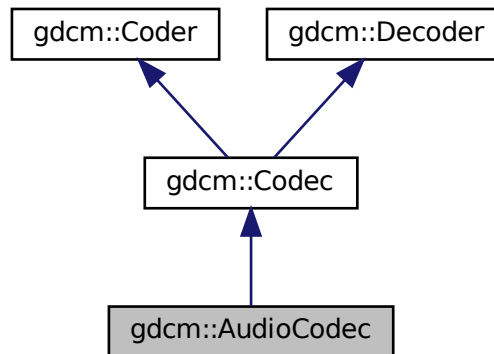
[AudioCodec.](#)

```
#include <gdcMAudioCodec.h>
```

Inheritance diagram for gdcM::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



## Public Member Functions

- [AudioCodec](#) ()
- [~AudioCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*

## Additional Inherited Members

### 10.26.1 Detailed Description

[AudioCodec](#).

### 10.26.2 Constructor & Destructor Documentation

#### 10.26.2.1 AudioCodec()

```
gdcm::AudioCodec::AudioCodec ( )
```

### 10.26.2.2 ~AudioCodec()

```
gdcM::AudioCodec::~~AudioCodec ( )
```

## 10.26.3 Member Function Documentation

### 10.26.3.1 CanCode()

```
bool gdcM::AudioCodec::CanCode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcM::Coder](#).

### 10.26.3.2 CanDecode()

```
bool gdcM::AudioCodec::CanDecode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcM::Decoder](#).

### 10.26.3.3 Decode()

```
bool gdcM::AudioCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcM::Decoder](#).

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

- [gdcMAudioCodec.h](#)

## 10.27 gdcm::Base64 Class Reference

Class for [Base64](#).

```
#include <gdcmBase64.h>
```

### 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)

### 10.27.1 Detailed Description

Class for [Base64](#).

### 10.27.2 Member Function Documentation

#### 10.27.2.1 Decode()

```
static size_t gdcm::Base64::Decode (
    char * dst,
    size_t dlen,
    const char * src,
    size_t slen ) [static]
```

Decode a base64-formatted buffer.

#### Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be decoded

#### Returns

0 if not successful, size of decoded otherwise

Examples:

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).

#### 10.27.2.2 Encode()

```
static size_t gdcM::Base64::Encode (
    char * dst,
    size_t dlen,
    const char * src,
    size_t slen ) [static]
```

Encode a buffer into base64 format.

##### Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be encoded

##### Returns

0 if not successful, size of encoded otherwise

#### 10.27.2.3 GetDecodeLength()

```
static size_t gdcM::Base64::GetDecodeLength (
    const char * src,
    size_t len ) [static]
```

Call this function to obtain the required buffer size

Examples:

[DumpExamCard.cxx](#), and [DumpSiemensBase64.cxx](#).



## 10.27.2.4 GetEncodeLength()

```
static size_t gdcm::Base64::GetEncodeLength (
    const char * src,
    size_t srclen ) [static]
```

Call this function to obtain the required buffer size

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

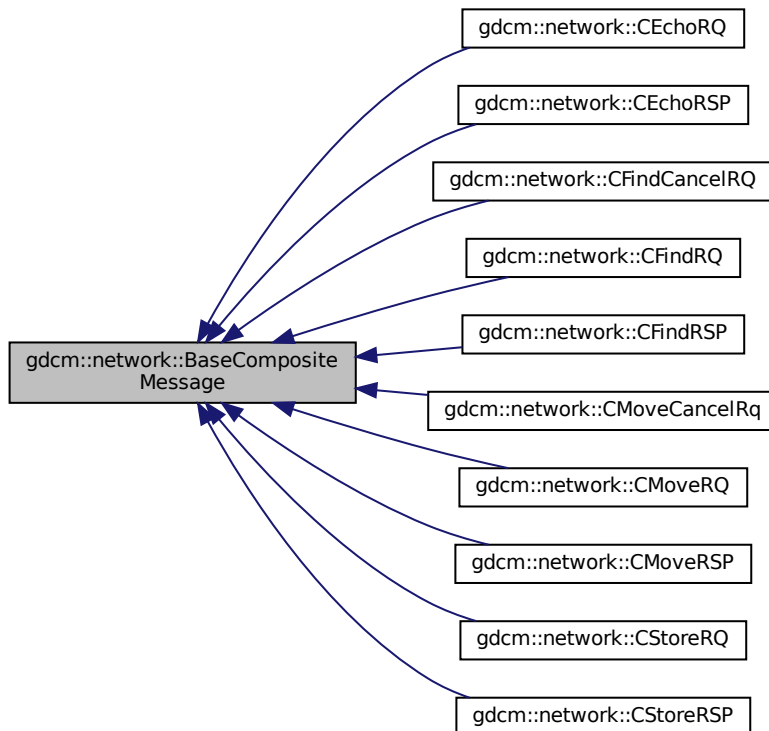
- [gdcmBase64.h](#)

## 10.28 gdcm::network::BaseCompositeMessage Class Reference

[BaseCompositeMessage](#).

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for gdcm::network::BaseCompositeMessage:



## Public Member Functions

- virtual `~BaseCompositeMessage()`
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseCompositeMessage *inRootQuery)=0`

### 10.28.1 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, `gdcmCompositeMessageFactory`.

This is an abstract class. It cannot be instantiated on its own.

### 10.28.2 Constructor & Destructor Documentation

#### 10.28.2.1 `~BaseCompositeMessage()`

```
virtual gdcm::network::BaseCompositeMessage::~~BaseCompositeMessage ( ) [inline], [virtual]
```

References `ConstructPDV()`.

### 10.28.3 Member Function Documentation

## 10.28.3.1 ConstructPDV()

```
virtual std::vector<PresentationDataValue> gdcm::network::BaseCompositeMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [pure virtual]
```

Implemented in [gdcm::network::CMoveRQ](#), [gdcm::network::CFindRQ](#), and [gdcm::network::CEchoRQ](#).

Referenced by `~BaseCompositeMessage()`.

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

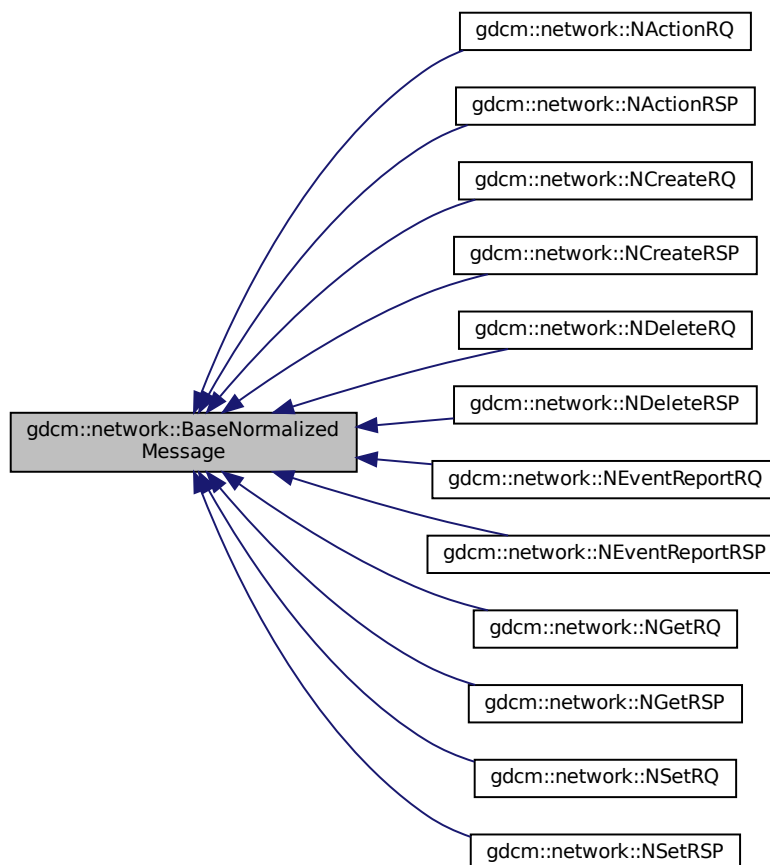
- [gdcmBaseCompositeMessage.h](#)

## 10.29 gdcm::network::BaseNormalizedMessage Class Reference

[BaseNormalizedMessage](#).

```
#include <gdcmBaseNormalizedMessage.h>
```

Inheritance diagram for `gdcm::network::BaseNormalizedMessage`:



## Public Member Functions

- virtual [~BaseNormalizedMessage](#) ()
- virtual std::vector< [PresentationDataValue](#) > [ConstructPDV](#) (const [ULConnection](#) &inConnection, const [BaseNormalizedMessage](#) \*inQuery)=0

### 10.29.1 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 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, [gdcmNormalizedMessageFactory.h](#).

This is an abstract class. It cannot be instantiated on its own.

### 10.29.2 Constructor & Destructor Documentation

#### 10.29.2.1 [~BaseNormalizedMessage](#)()

```
virtual gdcm::network::BaseNormalizedMessage::~~BaseNormalizedMessage ( ) [inline], [virtual]
```

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

### 10.29.3 Member Function Documentation

## 10.29.3.1 ConstructPDV()

```
virtual std::vector<PresentationDataValue> gdcmm::network::BaseNormalizedMessage::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [pure virtual]
```

Implemented in [gdcmm::network::NActionRQ](#), [gdcmm::network::NCreateRQ](#), [gdcmm::network::NDeleteRQ](#), [gdcmm::network::NEventReportRQ](#), [gdcmm::network::NGetRQ](#), and [gdcmm::network::NSetRQ](#).

Referenced by [~BaseNormalizedMessage\(\)](#).

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

- [gdcmmBaseNormalizedMessage.h](#)

## 10.30 gdcmm::network::BasePDU Class Reference

[BasePDU](#).

```
#include <gdcmmBasePDU.h>
```

Inheritance diagram for gdcmm::network::BasePDU:



## Public Member Functions

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

### 10.30.1 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

### 10.30.2 Constructor & Destructor Documentation

#### 10.30.2.1 [~BasePDU\(\)](#)

```
virtual gdcn::network::BasePDU::~~BasePDU ( ) [inline], [virtual]
```

References [IsLastFragment\(\)](#), [Print\(\)](#), [Read\(\)](#), [Size\(\)](#), and [Write\(\)](#).

### 10.30.3 Member Function Documentation

### 10.30.3.1 IsLastFragment()

```
virtual bool gdcmm::network::BasePDU::IsLastFragment ( ) const [pure virtual]
```

Implemented in [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::PDataTFPDU](#), [gdcmm::network::AAAbortPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AReleaseRPPDU](#), and [gdcmm::network::AReleaseRQPDU](#).

Referenced by [~BasePDU\(\)](#).

### 10.30.3.2 Print()

```
virtual void gdcmm::network::BasePDU::Print (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::PDataTFPDU](#), [gdcmm::network::AAAbortPDU](#), [gdcmm::network::AReleaseRPPDU](#), [gdcmm::network::AReleaseRQPDU](#), and [gdcmm::network::AAssociateRJPDU](#).

Referenced by [~BasePDU\(\)](#).

### 10.30.3.3 Read()

```
virtual std::istream& gdcmm::network::BasePDU::Read (
    std::istream & is ) [pure virtual]
```

Implemented in [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::PDataTFPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AReleaseRPPDU](#), [gdcmm::network::AReleaseRQPDU](#), and [gdcmm::network::AAAbortPDU](#).

Referenced by [~BasePDU\(\)](#).

### 10.30.3.4 Size()

```
virtual size_t gdcmm::network::BasePDU::Size ( ) const [pure virtual]
```

Implemented in [gdcmm::network::AAssociateACPDU](#), [gdcmm::network::AAssociateRQPDU](#), [gdcmm::network::PDataTFPDU](#), [gdcmm::network::AAAbortPDU](#), [gdcmm::network::AAssociateRJPDU](#), [gdcmm::network::AReleaseRPPDU](#), and [gdcmm::network::AReleaseRQPDU](#).

Referenced by [~BasePDU\(\)](#).

### 10.30.3.5 Write()

```
virtual const std::ostream& gdcm::network::BasePDU::Write (
    std::ostream & os ) const [pure virtual]
```

Implemented in [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::AAabortPDU](#).

Referenced by [~BasePDU\(\)](#).

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

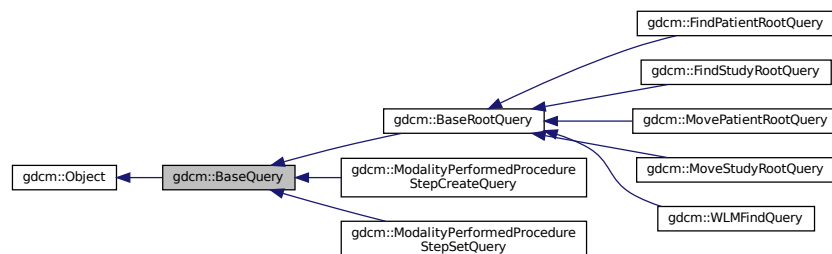
- [gdcmBasePDU.h](#)

## 10.31 gdcm::BaseQuery Class Reference

[BaseQuery](#).

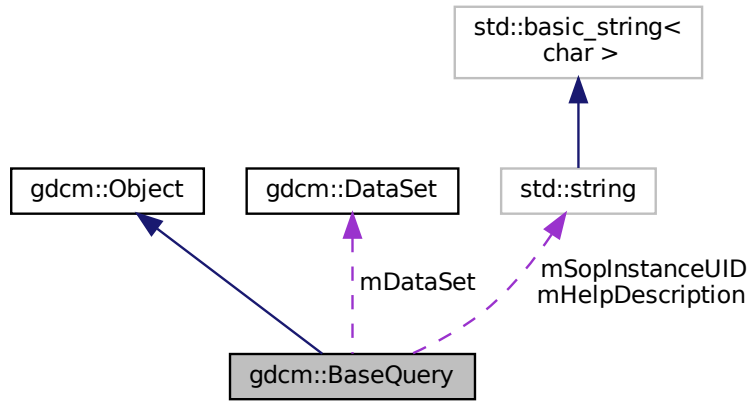
```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for [gdcm::BaseQuery](#):





Collaboration diagram for gdcm::BaseQuery:



## Public Member Functions

- virtual `~BaseQuery` ()
- void `AddQueryDataSet` (const `DataSet` &ds)
- virtual `UIDs::TSName GetAbstractSyntaxUID` () const =0
- `DataSet` const & `GetQueryDataSet` () const  
*Set/Get the internal representation of the query as a `DataSet`.*
- `DataSet` & `GetQueryDataSet` ()
- std::string `GetSOPInstanceUID` () const
- void `Print` (std::ostream &os) const
- void `SetSearchParameter` (const `Tag` &inTag, const std::string &inValue)
- void `SetSearchParameter` (const std::string &inKeyword, 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)

## 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 Attributes

- `DataSet` `mDataSet`
- std::string `mHelpDescription`
- std::string `mSopInstanceUID`

## Friends

- class [QueryFactory](#)

### 10.31.1 Detailed Description

[BaseQuery](#).

contains: a baseclass which will produce a dataset for all dimse messages

### 10.31.2 Constructor & Destructor Documentation

#### 10.31.2.1 BaseQuery()

```
gdcm::BaseQuery::BaseQuery ( ) [protected]
```

#### 10.31.2.2 ~BaseQuery()

```
virtual gdcm::BaseQuery::~~BaseQuery ( ) [virtual]
```

### 10.31.3 Member Function Documentation

#### 10.31.3.1 AddQueryDataSet()

```
void gdcm::BaseQuery::AddQueryDataSet (
    const DataSet & ds )
```

#### 10.31.3.2 GetAbstractSyntaxUID()

```
virtual UIDs::TSName gdcm::BaseQuery::GetAbstractSyntaxUID ( ) const [pure virtual]
```

Implemented in [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), [gdcm::WLMFindQuery](#), [gdcm::FindPatientRootQuery](#), [gdcm::ModalityPerformedProcedureStepCreateQuery](#), and [gdcm::ModalityPerformedProcedureStepSetQuery](#).

### 10.31.3.3 GetQueryDataSet() [1/2]

```
DataSet const& gdcm::BaseQuery::GetQueryDataSet ( ) const
```

Set/Get the internal representation of the query as a [DataSet](#).

### 10.31.3.4 GetQueryDataSet() [2/2]

```
DataSet& gdcm::BaseQuery::GetQueryDataSet ( )
```

### 10.31.3.5 GetSOPInstanceUID()

```
std::string gdcm::BaseQuery::GetSOPInstanceUID ( ) const [inline]
```

### 10.31.3.6 Print()

```
void gdcm::BaseQuery::Print (
    std::ostream & os ) const [virtual]
```

Reimplemented from [gdcm::Object](#).

### 10.31.3.7 SetSearchParameter() [1/3]

```
void gdcm::BaseQuery::SetSearchParameter (
    const Tag & inTag,
    const DictEntry & inDictEntry,
    const std::string & inValue ) [protected]
```

### 10.31.3.8 SetSearchParameter() [2/3]

```
void gdcm::BaseQuery::SetSearchParameter (
    const Tag & inTag,
    const std::string & inValue )
```

### 10.31.3.9 SetSearchParameter() [3/3]

```
void gdcM::BaseQuery::SetSearchParameter (
    const std::string & inKeyword,
    const std::string & inValue )
```

### 10.31.3.10 SetSOPInstanceUID()

```
void gdcM::BaseQuery::SetSOPInstanceUID (
    const std::string & iSopInstanceUID ) [inline]
```

### 10.31.3.11 ValidateQuery()

```
virtual bool gdcM::BaseQuery::ValidateQuery (
    bool inStrict = true ) const [pure virtual]
```

Implemented in [gdcM::BaseRootQuery](#), [gdcM::FindStudyRootQuery](#), [gdcM::MovePatientRootQuery](#), [gdcM::MoveStudyRootQuery](#), [gdcM::WLMFindQuery](#), [gdcM::FindPatientRootQuery](#), [gdcM::ModalityPerformedProcedureStepCreateQuery](#), and [gdcM::ModalityPerformedProcedureStepSetQuery](#).

### 10.31.3.12 ValidDataSet()

```
bool gdcM::BaseQuery::ValidDataSet (
    const DataSet & dataSetToValid,
    const DataSet & dataSetReference ) const [protected]
```

### 10.31.3.13 WriteHelpFile()

```
const std::ostream& gdcM::BaseQuery::WriteHelpFile (
    std::ostream & os )
```

### 10.31.3.14 WriteQuery()

```
bool gdcM::BaseQuery::WriteQuery (
    const std::string & inFileName )
```

## 10.31.4 Friends And Related Function Documentation

### 10.31.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

## 10.31.5 Member Data Documentation

### 10.31.5.1 mDataSet

```
DataSet gdcm::BaseQuery::mDataSet [protected]
```

### 10.31.5.2 mHelpDescription

```
std::string gdcm::BaseQuery::mHelpDescription [protected]
```

### 10.31.5.3 mSopInstanceUID

```
std::string gdcm::BaseQuery::mSopInstanceUID [protected]
```

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

- [gdcmBaseQuery.h](#)

## 10.32 gdcm::BaseRootQuery Class Reference

[BaseRootQuery](#).

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



### Public Member Functions

- virtual [~BaseRootQuery](#) ()
- [EQueryLevel](#) [GetQueryLevelFromQueryRoot](#) ([ERootType](#) roottype)
- virtual std::vector< [Tag](#) > [GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)=0
- virtual void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)=0
- virtual bool [ValidateQuery](#) (bool inStrict=true) const =0

### Static Public Member Functions

- static [QueryBase](#) \* [Construct](#) ([ERootType](#) inRootType, [EQueryLevel](#) qllevel)
- static int [GetQueryLevelFromString](#) (const char \*str)
- static const char \* [GetQueryLevelString](#) ([EQueryLevel](#) ql)

## Protected Member Functions

- [BaseRootQuery \(\)](#)

## Protected Attributes

- `std::string` [mHelpDescription](#)
- [QueryImage](#) [mImage](#)
- [QueryPatient](#) [mPatient](#)
- [ERootType](#) [mRootType](#)
- [QuerySeries](#) [mSeries](#)
- [QueryStudy](#) [mStudy](#)

## Friends

- class [QueryFactory](#)

### 10.32.1 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.

### 10.32.2 Constructor & Destructor Documentation

#### 10.32.2.1 [BaseRootQuery\(\)](#)

```
gdcm::BaseRootQuery::BaseRootQuery ( ) [protected]
```

#### 10.32.2.2 [~BaseRootQuery\(\)](#)

```
virtual gdcm::BaseRootQuery::~BaseRootQuery ( ) [virtual]
```

### 10.32.3 Member Function Documentation

#### 10.32.3.1 Construct()

```
static QueryBase* gdcm::BaseRootQuery::Construct (
    ERootType inRootType,
    EQueryLevel qllevel ) [static]
```

#### 10.32.3.2 GetQueryLevelFromQueryRoot()

```
EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (
    ERootType roottype )
```

#### 10.32.3.3 GetQueryLevelFromString()

```
static int gdcm::BaseRootQuery::GetQueryLevelFromString (
    const char * str ) [static]
```

#### 10.32.3.4 GetQueryLevelString()

```
static const char* gdcm::BaseRootQuery::GetQueryLevelString (
    EQueryLevel ql ) [static]
```

#### 10.32.3.5 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](#).



### 10.32.3.6 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 dcm4k

Implemented in [gdcm::WLMFindQuery](#), [gdcm::FindPatientRootQuery](#), [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), and [gdcm::MoveStudyRootQuery](#).

### 10.32.3.7 ValidateQuery()

```
virtual bool gdcm::BaseRootQuery::ValidateQuery (
    bool inStrict = true ) const [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 [gdcm::BaseQuery](#).

Implemented in [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), [gdcm::WLMFindQuery](#), and [gdcm::FindPatientRootQuery](#).

## 10.32.4 Friends And Related Function Documentation

### 10.32.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

## 10.32.5 Member Data Documentation

#### 10.32.5.1 mHelpDescription

`std::string gdcM::BaseRootQuery::mHelpDescription [protected]`

#### 10.32.5.2 mImage

`QueryImage gdcM::BaseRootQuery::mImage [protected]`

#### 10.32.5.3 mPatient

`QueryPatient gdcM::BaseRootQuery::mPatient [protected]`

#### 10.32.5.4 mRootType

`ERootType gdcM::BaseRootQuery::mRootType [protected]`

#### 10.32.5.5 mSeries

`QuerySeries gdcM::BaseRootQuery::mSeries [protected]`

#### 10.32.5.6 mStudy

`QueryStudy gdcM::BaseRootQuery::mStudy [protected]`

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

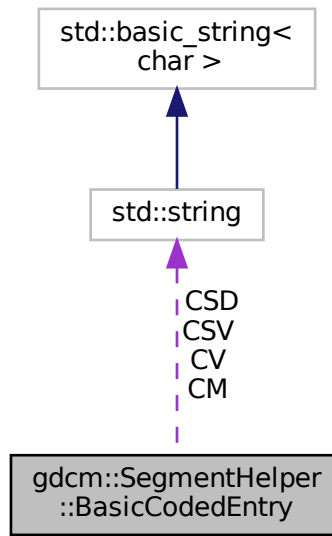
- [gdcMBaseRootQuery.h](#)

## 10.33 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](#)

### 10.33.1 Detailed Description

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

See also

PS 3.3 section 8.8.

### 10.33.2 Constructor & Destructor Documentation

#### 10.33.2.1 BasicCodedEntry() [1/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]
```

Constructor.

#### 10.33.2.2 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.

#### 10.33.2.3 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.

### 10.33.3 Member Function Documentation

#### 10.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
    const bool checkOptionalAttributes = false ) const
```

Check if each attributes of the basic coded entry is defined.

## Parameters

<i>checkOptionalAttributes</i>	Check also type 1C attributes.
--------------------------------	--------------------------------

## 10.33.4 Member Data Documentation

### 10.33.4.1 CM

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CM
```

Coding Scheme [Version](#) attribute.

### 10.33.4.2 CSD

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSD
```

Code [Value](#) attribute.

### 10.33.4.3 CSV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSV
```

Coding Scheme Designator attribute.

### 10.33.4.4 CV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CV
```

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

- [gdcmSegmentHelper.h](#)

## 10.34 gdcm::BasicOffsetTable Class Reference

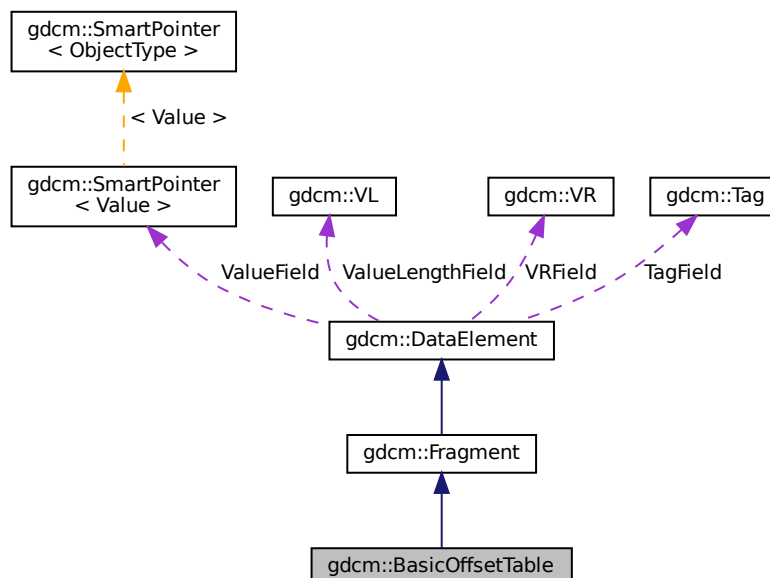
Class to represent a [BasicOffsetTable](#).

```
#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)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [BasicOffsetTable](#) &val)

## Additional Inherited Members

### 10.34.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

### 10.34.2 Constructor & Destructor Documentation

#### 10.34.2.1 BasicOffsetTable()

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

References [gdcm::operator<<\(\)](#).

### 10.34.3 Member Function Documentation

#### 10.34.3.1 Read()

```
template<typename TSwap >  
std::istream& gdcm::BasicOffsetTable::Read (  
    std::istream & is ) [inline]
```

References [gdcmDebugMacro](#).

### 10.34.4 Friends And Related Function Documentation

#### 10.34.4.1 operator<<

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

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

- [gdcmBasicOffsetTable.h](#)

### 10.35 gdcm::Bitmap Class Reference

[Bitmap](#) class.

```
#include <gdcmBitmap.h>
```

Inheritance diagram for gdcm::Bitmap:





Collaboration diagram for gdcm::Bitmap:



## Public Member Functions

- [Bitmap](#) ()
- [~Bitmap](#) ()
- virtual bool [AreOverlaysInPixelData](#) () const
- void [Clear](#) ()
- bool [GetBuffer](#) (char \*buffer) const  
*Acces the raw data.*
- unsigned long [GetBufferLength](#) () const
- unsigned int [GetColumns](#) () const
- const [DataElement](#) & [GetDataElement](#) () const
- [DataElement](#) & [GetDataElement](#) ()
- 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)...*
- const [LookupTable](#) & [GetLUT](#) () const
- [LookupTable](#) & [GetLUT](#) ()
- bool [GetNeedByteSwap](#) () const
- 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*
- const [PixelFormat](#) & [GetPixelFormat](#) () const  
*Get/Set PixelFormat.*
- [PixelFormat](#) & [GetPixelFormat](#) ()
- 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

- 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

## 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 [TryKAKADUCoec](#) (char \*buffer, bool &lossyflag) const
- bool [TryPVRGCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRAWCodec](#) (char \*buffer, bool &lossyflag) const
- bool [TryRLECoec](#) (char \*buffer, bool &lossyflag) const

## 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](#)

### 10.35.1 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)

Examples:

[ExtractIconFromFile.cxx](#).

### 10.35.2 Member Typedef Documentation

#### 10.35.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]
```

### 10.35.3 Constructor & Destructor Documentation

#### 10.35.3.1 Bitmap()

```
gdcm::Bitmap::Bitmap ( )
```

#### 10.35.3.2 ~Bitmap()

```
gdcm::Bitmap::~~Bitmap ( )
```

### 10.35.4 Member Function Documentation

#### 10.35.4.1 AreOverlaysInPixelData()

```
virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const [inline], [virtual]
```

Reimplemented in [gdcm::Pixmap](#).

#### 10.35.4.2 Clear()

```
void gdcm::Bitmap::Clear ( )
```

#### 10.35.4.3 ComputeLossyFlag()

```
bool gdcm::Bitmap::ComputeLossyFlag ( ) [protected]
```

#### 10.35.4.4 GetBuffer()

```
bool gdcm::Bitmap::GetBuffer (
    char * buffer ) const
```

Access the raw data.

#### Examples:

[ConvertToQImage.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.5 GetBuffer2()

```
bool gdcm::Bitmap::GetBuffer2 (
    std::ostream & os ) const [protected]
```

#### 10.35.4.6 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:

[ConvertToQImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.7 GetColumns()

```
unsigned int gdcm::Bitmap::GetColumns ( ) const [inline]
```

#### 10.35.4.8 GetDataElement() [1/2]

```
const DataElement& gdcm::Bitmap::GetDataElement ( ) const [inline]
```

##### Examples:

[ExtractIconFromFile.cxx](#).

#### 10.35.4.9 GetDataElement() [2/2]

```
DataElement& gdcm::Bitmap::GetDataElement ( ) [inline]
```

#### 10.35.4.10 GetDimension()

```
unsigned int gdcm::Bitmap::GetDimension (
    unsigned int idx ) const
```

#### 10.35.4.11 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](#).↵

#### 10.35.4.12 GetLUT() [1/2]

```
const LookupTable& gdcm::Bitmap::GetLUT ( ) const [inline]
```

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.35.4.13 GetLUT() [2/2]

```
LookupTable& gdcm::Bitmap::GetLUT ( ) [inline]
```

#### 10.35.4.14 GetNeedByteSwap()

```
bool gdcm::Bitmap::GetNeedByteSwap ( ) const [inline]
```

#### 10.35.4.15 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:

[HelloVizWorld.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const
```

return the photometric interpretation

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

#### 10.35.4.17 GetPixelFormat() [1/2]

```
const PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) const [inline]
```

Get/Set [PixelFormat](#).

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

#### 10.35.4.18 GetPixelFormat() [2/2]

```
PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) [inline]
```

#### 10.35.4.19 GetPlanarConfiguration()

```
unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const
```

return the planar configuration

#### 10.35.4.20 GetRows()

```
unsigned int gdcm::Bitmap::GetRows ( ) const [inline]
```

#### 10.35.4.21 GetTransferSyntax()

```
const TransferSyntax& gdcm::Bitmap::GetTransferSyntax ( ) const [inline]
```

##### Examples:

[ExtractIconFromFile.cxx](#).

#### 10.35.4.22 IsEmpty()

```
bool gdcm::Bitmap::IsEmpty ( ) const [inline]
```

#### 10.35.4.23 IsLossy()

```
bool gdcm::Bitmap::IsLossy ( ) const
```

Return whether or not the image was compressed using a lossy compressor or not.

#### 10.35.4.24 IsTransferSyntaxCompatible()

```
bool gdcm::Bitmap::IsTransferSyntaxCompatible (
    TransferSyntax const & ts ) const
```

#### 10.35.4.25 Print()

```
void gdcm::Bitmap::Print (
    std::ostream & ) const [virtual]
```

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Image](#), and [gdcm::Pixmap](#).

##### Examples:

[ExtractIconFromFile.cxx](#).



#### 10.35.4.26 SetColumns()

```
void gdcm::Bitmap::SetColumns (
    unsigned int col ) [inline]
```

#### 10.35.4.27 SetDataElement()

```
void gdcm::Bitmap::SetDataElement (
    DataElement const & de ) [inline]
```

##### Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
    unsigned int idx,
    unsigned int dim )
```

##### Examples:

[csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
    const unsigned int dims[3] )
```

##### Examples:

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

#### 10.35.4.30 SetLossyFlag()

```
void gdcm::Bitmap::SetLossyFlag (
    bool f ) [inline]
```

Specifically set that the image was compressed using a lossy compression mechanism.

#### 10.35.4.31 SetLUT()

```
void gdcm::Bitmap::SetLUT (
    LookupTable const & lut ) [inline]
```

Set/Get LUT.

#### 10.35.4.32 SetNeedByteSwap()

```
void gdcm::Bitmap::SetNeedByteSwap (
    bool b ) [inline]
```

#### 10.35.4.33 SetNumberOfDimensions()

```
void gdcm::Bitmap::SetNumberOfDimensions (
    unsigned int dim )
```

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [FileChangeTS.cs](#), [FileChangeTSLossy.cs](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

#### 10.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

##### Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References [gdcm::PixelFormat::Validate\(\)](#).

#### 10.35.4.36 SetPlanarConfiguration()

```
void gdcm::Bitmap::SetPlanarConfiguration (
    unsigned int pc )
```

##### Warning

you need to call [SetPixelFormat](#) first (before [SetPlanarConfiguration](#)) for consistency checking

#### 10.35.4.37 SetRows()

```
void gdcm::Bitmap::SetRows (
    unsigned int rows ) [inline]
```

#### 10.35.4.38 SetTransferSyntax()

```
void gdcm::Bitmap::SetTransferSyntax (
    TransferSyntax const & ts ) [inline]
```

Transfer syntax.

##### Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [MergeTwoFiles.cxx](#).

#### 10.35.4.39 TryJPEG2000Codec()

```
bool gdcm::Bitmap::TryJPEG2000Codec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.40 TryJPEG2000Codec2()

```
bool gdcm::Bitmap::TryJPEG2000Codec2 (
    std::ostream & os ) const    [protected]
```

#### 10.35.4.41 TryJPEGCodec()

```
bool gdcm::Bitmap::TryJPEGCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.42 TryJPEGCodec2()

```
bool gdcm::Bitmap::TryJPEGCodec2 (
    std::ostream & os ) const    [protected]
```

#### 10.35.4.43 TryJPEGLSCodec()

```
bool gdcm::Bitmap::TryJPEGLSCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.44 TryKAKADUCodec()

```
bool gdcm::Bitmap::TryKAKADUCodec (
    char * buffer,
    bool & lossyflag ) const    [protected]
```

#### 10.35.4.45 TryPVRGCodec()

```
bool gdcm::Bitmap::TryPVRGCodec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.46 TryRAWCodec()

```
bool gdcm::Bitmap::TryRAWCodec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.47 TryRLECodec()

```
bool gdcm::Bitmap::TryRLECodec (
    char * buffer,
    bool & lossyflag ) const [protected]
```

#### 10.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcm::Bitmap::UnusedBitsPresentInPixelData ( ) const [inline], [virtual]
```

Reimplemented in [gdcm::Pixmap](#).

References [gdcm::terminal::dim](#).

### 10.35.5 Friends And Related Function Documentation

#### 10.35.5.1 ImageChangeTransferSyntax

```
friend class ImageChangeTransferSyntax [friend]
```

### 10.35.5.2 PixmapReader

```
friend class PixmapReader [friend]
```

## 10.35.6 Member Data Documentation

### 10.35.6.1 Dimensions

```
std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]
```

### 10.35.6.2 LossyFlag

```
bool gdcm::Bitmap::LossyFlag [protected]
```

### 10.35.6.3 LUT

```
LUTPtr gdcm::Bitmap::LUT [protected]
```

### 10.35.6.4 NeedByteSwap

```
bool gdcm::Bitmap::NeedByteSwap [protected]
```

### 10.35.6.5 NumberOfDimensions

```
unsigned int gdcm::Bitmap::NumberOfDimensions [protected]
```

### 10.35.6.6 PF

```
PixelFormat gdcm::Bitmap::PF [protected]
```

## 10.35.6.7 PI

`PhotometricInterpretation` gdcm::Bitmap::PI [protected]

## 10.35.6.8 PixelData

`DataElement` gdcm::Bitmap::PixelData [protected]

## 10.35.6.9 PlanarConfiguration

`unsigned int` gdcm::Bitmap::PlanarConfiguration [protected]

## 10.35.6.10 TS

`TransferSyntax` gdcm::Bitmap::TS [protected]

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

- [gdcmBitmap.h](#)

## 10.36 gdcm::BitmapToBitmapFilter Class Reference

`BitmapToBitmapFilter` class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for `gdcm::BitmapToBitmapFilter`:



## Public Member Functions

- [BitmapToBitmapFilter \(\)](#)
- [~BitmapToBitmapFilter \(\)](#)
- `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](#)

### 10.36.1 Detailed Description

[BitmapToBitmapFilter](#) class.

Super class for all filter taking an image and producing an output image



## 10.36.2 Constructor & Destructor Documentation

### 10.36.2.1 BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )
```

### 10.36.2.2 ~BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ( ) [inline]
```

## 10.36.3 Member Function Documentation

### 10.36.3.1 GetOutput()

```
const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput ( ) const [inline]
```

Get Output image.

### 10.36.3.2 GetOutputAsBitmap()

```
const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const
```

### 10.36.3.3 SetInput()

```
void gdcm::BitmapToBitmapFilter::SetInput (
    const Bitmap & image )
```

Set input image.

Examples:

[CompressImage.cxx](#).

## 10.36.4 Member Data Documentation

### 10.36.4.1 Input

```
SmartPointer<Bitmap> gdcM::BitmapToBitmapFilter::Input [protected]
```

### 10.36.4.2 Output

```
SmartPointer<Bitmap> gdcM::BitmapToBitmapFilter::Output [protected]
```

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

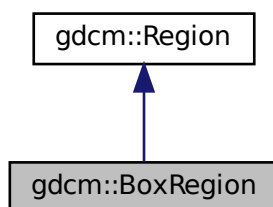
- [gdcMBitmapToBitmapFilter.h](#)

## 10.37 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](#) ()
- [size\\_t Area](#) () const  
*compute the area*
- [Region \\* Clone](#) () const
- [BoxRegion ComputeBoundingBox](#) ()  
*Return the Axis-Aligned minimum bounding box for all regions.*
- [bool Empty](#) () const  
*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  
*return whether this is valid domain*
- [void operator=](#) (const [BoxRegion](#) &)
- [void Print](#) (std::ostream &os=std::cout) const  
*Print.*
- [void SetDomain](#) (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)  
*Set domain.*

### Static Public Member Functions

- static [BoxRegion BoundingBox](#) ([BoxRegion](#) const &b1, [BoxRegion](#) const &b2)  
*Helper class to compute the bounding box of two [BoxRegion](#).*

### 10.37.1 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)

### 10.37.2 Constructor & Destructor Documentation

#### 10.37.2.1 [BoxRegion\(\)](#) [1/2]

```
gdcm::BoxRegion::BoxRegion ( )
```

#### 10.37.2.2 [~BoxRegion\(\)](#)

```
gdcm::BoxRegion::~~BoxRegion ( )
```

#### 10.37.2.3 [BoxRegion\(\)](#) [2/2]

```
gdcm::BoxRegion::BoxRegion (
    const BoxRegion & )
```

copy/cstor and al.

### 10.37.3 Member Function Documentation

#### 10.37.3.1 [Area\(\)](#)

```
size_t gdcm::BoxRegion::Area ( ) const [virtual]
```

compute the area

Implements [gdcm::Region](#).

### 10.37.3.2 BoundingBox()

```
static BoxRegion gdcm::BoxRegion::BoundingBox (
    BoxRegion const & b1,
    BoxRegion const & b2 ) [static]
```

Helper class to compute the bounding box of two [BoxRegion](#).

### 10.37.3.3 Clone()

```
Region* gdcm::BoxRegion::Clone ( ) const [virtual]
```

Implements [gdcm::Region](#).

### 10.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( ) [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcm::Region](#).

### 10.37.3.5 Empty()

```
bool gdcm::BoxRegion::Empty ( ) const [virtual]
```

return whether this domain is empty:

Implements [gdcm::Region](#).

### 10.37.3.6 GetXMax()

```
unsigned int gdcm::BoxRegion::GetXMax ( ) const
```

#### 10.37.3.7 GetXMin()

```
unsigned int gdcm::BoxRegion::GetXMin ( ) const
```

Get domain.

#### 10.37.3.8 GetYMax()

```
unsigned int gdcm::BoxRegion::GetYMax ( ) const
```

#### 10.37.3.9 GetYMin()

```
unsigned int gdcm::BoxRegion::GetYMin ( ) const
```

#### 10.37.3.10 GetZMax()

```
unsigned int gdcm::BoxRegion::GetZMax ( ) const
```

#### 10.37.3.11 GetZMin()

```
unsigned int gdcm::BoxRegion::GetZMin ( ) const
```

#### 10.37.3.12 IsValid()

```
bool gdcm::BoxRegion::IsValid ( ) const [virtual]
```

return whether this is valid domain

Implements [gdcm::Region](#).

### 10.37.3.13 operator=()

```
void gdcm::BoxRegion::operator= (
    const BoxRegion & )
```

### 10.37.3.14 Print()

```
void gdcm::BoxRegion::Print (
    std::ostream & os = std::cout ) const [virtual]
```

Print.

Reimplemented from [gdcm::Region](#).

### 10.37.3.15 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.

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

- [gdcmBoxRegion.h](#)

## 10.38 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](#) ()

### 10.38.1 Detailed Description

[ByteBuffer](#).

Detailed description here

#### Note

looks like a `std::streambuf` or `std::filebuf` class with the get and peek pointer

### 10.38.2 Constructor & Destructor Documentation

#### 10.38.2.1 ByteBuffer()

```
gdcM::ByteBuffer::ByteBuffer ( ) [inline]
```

### 10.38.3 Member Function Documentation

#### 10.38.3.1 Get()

```
char* gdcM::ByteBuffer::Get (
    int len ) [inline]
```

#### 10.38.3.2 GetStart()

```
const char* gdcM::ByteBuffer::GetStart ( ) const [inline]
```

#### 10.38.3.3 ShiftEnd()

```
void gdcM::ByteBuffer::ShiftEnd (
    int len ) [inline]
```



#### 10.38.3.4 UpdatePosition()

```
void gdcm::ByteBuffer::UpdatePosition ( ) [inline]
```

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

- [gdcmByteBuffer.h](#)

## 10.39 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](#) ()

### 10.39.1 Detailed Description

```
template<class T>  
class gdcm::ByteSwap< T >
```

[ByteSwap.](#)

Perform machine dependent byte swaping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap\_32 / bswap\_64 ...

Examples:

[TestByteSwap.cxx.](#)

### 10.39.2 Member Function Documentation

### 10.39.2.1 Swap()

```
template<class T >
static void gdcm::ByteSwap< T >::Swap (
    T & p ) [static]
```

### 10.39.2.2 SwapFromSwapCodeIntoSystem()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (
    T & p,
    SwapCode const & sc ) [static]
```

#### Examples:

[TestByteSwap.cxx](#).

### 10.39.2.3 SwapRange()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapRange (
    T * p,
    unsigned int num ) [static]
```

### 10.39.2.4 SwapRangeFromSwapCodeIntoSystem()

```
template<class T >
static void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (
    T * p,
    SwapCode const & sc,
    std::streamoff num ) [static]
```

#### Examples:

[TestByteSwap.cxx](#).

#### 10.39.2.5 SystemIsBigEndian()

```
template<class T >
static bool gdcm::ByteSwap< T >::SystemIsBigEndian ( ) [static]
```

Query the machine Endian-ness.

#### 10.39.2.6 SystemIsLittleEndian()

```
template<class T >
static bool gdcm::ByteSwap< T >::SystemIsLittleEndian ( ) [static]
```

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

- [gdcmByteSwap.h](#)

## 10.40 gdcm::ByteSwapFilter Class Reference

[ByteSwapFilter](#).

```
#include <gdcmByteSwapFilter.h>
```

### Public Member Functions

- [ByteSwapFilter](#) ([DataSet](#) &ds)
- [~ByteSwapFilter](#) ()
- bool [ByteSwap](#) ()
- void [SetByteSwapTag](#) (bool b)

#### 10.40.1 Detailed Description

[ByteSwapFilter](#).

In place byte-swapping of a dataset FIXME: FL status ??

#### 10.40.2 Constructor & Destructor Documentation

#### 10.40.2.1 ByteSwapFilter()

```
gdcM::ByteSwapFilter::ByteSwapFilter (
    DataSet & ds ) [inline]
```

#### 10.40.2.2 ~ByteSwapFilter()

```
gdcM::ByteSwapFilter::~~ByteSwapFilter ( )
```

### 10.40.3 Member Function Documentation

#### 10.40.3.1 ByteSwap()

```
bool gdcM::ByteSwapFilter::ByteSwap ( )
```

#### 10.40.3.2 SetByteSwapTag()

```
void gdcM::ByteSwapFilter::SetByteSwapTag (
    bool b ) [inline]
```

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

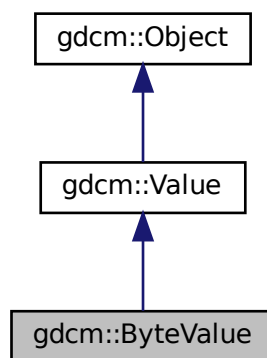
- [gdcMByteSwapFilter.h](#)

## 10.41 gdcm::ByteValue Class Reference

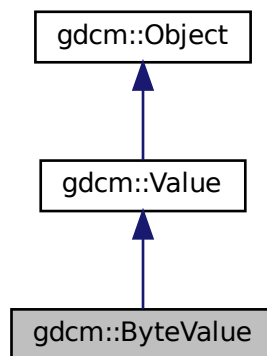
Class to represent binary value (array of bytes)

```
#include <gdcmByteValue.h>
```

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



## Public Member Functions

- [ByteValue](#) (const char \*array=0, [VL](#) const &vl=0)
- [ByteValue](#) (std::vector< char > &v)
- [~ByteValue](#) ()
- void [Append](#) ([ByteValue](#) const &bv)
- void [Clear](#) ()
- [VL ComputeLength](#) () const
- void [Fill](#) (char c)
- bool [GetBuffer](#) (char \*buffer, unsigned long length) const
- [VL GetLength](#) () const
- const char \* [GetPointer](#) () 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
- void [PrintASCII](#) (std::ostream &os, [VL](#) maxlength) const
- void [PrintASCIIXML](#) (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 , typename TType >  
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- void [SetLength](#) ([VL](#) vl)
- template<typename TSwap , typename TType >  
std::ostream const & [Write](#) (std::ostream &os) const
- template<typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const
- bool [WriteBuffer](#) (std::ostream &os) const

## Protected Member Functions

- void [Print](#) (std::ostream &os) const
- void [SetLengthOnly](#) ([VL](#) vl)

### 10.41.1 Detailed Description

Class to represent binary value (array of bytes)

## Note

## Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), and [rle2img.cxx](#).

## 10.41.2 Constructor & Destructor Documentation

### 10.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (
    const char * array = 0,
    VL const & vl = 0 ) [inline]
```

References [gdcmDebugMacro](#).

### 10.41.2.2 ByteValue() [2/2]

```
gdcm::ByteValue::ByteValue (
    std::vector< char > & v ) [inline]
```

## Warning

casting to `uint32_t`

### 10.41.2.3 ~ByteValue()

```
gdcm::ByteValue::~ByteValue ( ) [inline]
```

## 10.41.3 Member Function Documentation

#### 10.41.3.1 Append()

```
void gdcM::ByteValue::Append (
    ByteValue const & bv )
```

#### 10.41.3.2 Clear()

```
void gdcM::ByteValue::Clear ( ) [inline], [virtual]
```

Implements [gdcM::Value](#).

#### 10.41.3.3 ComputeLength()

```
VL gdcM::ByteValue::ComputeLength ( ) const [inline]
```

Referenced by [gdcM::Fragment::Write\(\)](#).

#### 10.41.3.4 Fill()

```
void gdcM::ByteValue::Fill (
    char c ) [inline]
```

Examples:

[DuplicatePCDE.cxx](#).

#### 10.41.3.5 GetBuffer()

```
bool gdcM::ByteValue::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

Examples:

[FixJAIBugJPEGLS.cxx](#).



## 10.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength ( ) const [inline], [virtual]
```

Implements [gdcm::Value](#).

## Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::operator<<\(\)](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< VR::OB, VM::VM1\\_n >::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\\_n >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap\(\)](#), [gdcm::Element< VR::OB, VM::VM1\\_n >::SetNoSwap\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::SetNoSwap\(\)](#), and [gdcm::Fragment::Write\(\)](#).

## 10.41.3.7 GetPointer()

```
const char* gdcm::ByteValue::GetPointer ( ) const [inline]
```

## Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::operator<<\(\)](#), [gdcm::SequenceOfFragments::ReadValue\(\)](#), [gdcm::Element< VR::OB, VM::VM1\\_n >::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\\_n >::SetByteValue\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap\(\)](#), [gdcm::Element< VR::OB, VM::VM1\\_n >::SetNoSwap\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::SetNoSwap\(\)](#).

## 10.41.3.8 IsEmpty()

```
bool gdcm::ByteValue::IsEmpty ( ) const [inline]
```

#### 10.41.3.9 IsPrintable()

```
bool gdcmm::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...

#### 10.41.3.10 operator const std::vector< char > &()

```
gdcmm::ByteValue::operator const std::vector< char > & ( ) const [inline]
```

#### 10.41.3.11 operator=()

```
ByteValue& gdcmm::ByteValue::operator= (
    const ByteValue & val ) [inline]
```

#### 10.41.3.12 operator==( ) [1/2]

```
bool gdcmm::ByteValue::operator== (
    const ByteValue & val ) const [inline]
```

#### 10.41.3.13 operator==( ) [2/2]

```
bool gdcmm::ByteValue::operator== (
    const Value & val ) const [inline], [virtual]
```

Implements [gdcmm::Value](#).

#### 10.41.3.14 Print()

```
void gdcmm::ByteValue::Print (
    std::ostream & os ) const [inline], [protected], [virtual]
```

Reimplemented from [gdcmm::Object](#).

#### 10.41.3.15 PrintASCII()

```
void gdcm::ByteValue::PrintASCII (
    std::ostream & os,
    VL maxlength ) const
```

#### 10.41.3.16 PrintASCIIXML()

```
void gdcm::ByteValue::PrintASCIIXML (
    std::ostream & os ) const
```

#### 10.41.3.17 PrintGroupLength()

```
void gdcm::ByteValue::PrintGroupLength (
    std::ostream & os ) [inline]
```

#### 10.41.3.18 PrintHex()

```
void gdcm::ByteValue::PrintHex (
    std::ostream & os,
    VL maxlength ) const
```

#### 10.41.3.19 PrintHexXML()

```
void gdcm::ByteValue::PrintHexXML (
    std::ostream & os ) const
```

#### 10.41.3.20 PrintPNXML()

```
void gdcm::ByteValue::PrintPNXML (
    std::ostream & os ) const
```

To Print Values in Native DICOM format

**10.41.3.21 Read()** [1/2]

```
template<typename TSwap , typename TType >
std::istream& gdcM::ByteValue::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

**10.41.3.22 Read()** [2/2]

```
template<typename TSwap >
std::istream& gdcM::ByteValue::Read (
    std::istream & is ) [inline]
```

**10.41.3.23 SetLength()**

```
void gdcM::ByteValue::SetLength (
    VL vl ) [virtual]
```

Implements [gdcM::Value](#).

**10.41.3.24 SetLengthOnly()**

```
void gdcM::ByteValue::SetLengthOnly (
    VL vl ) [inline], [protected], [virtual]
```

Reimplemented from [gdcM::Value](#).

**10.41.3.25 Write()** [1/2]

```
template<typename TSwap , typename TType >
std::ostream const& gdcM::ByteValue::Write (
    std::ostream & os ) const [inline]
```

Referenced by [gdcM::Fragment::Write\(\)](#).

### 10.41.3.26 Write() [2/2]

```
template<typename TSwap >
std::ostream const& gdcm::ByteValue::Write (
    std::ostream & os ) const [inline]
```

### 10.41.3.27 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (
    std::ostream & os ) const [inline]
```

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

- [gdcmByteValue.h](#)

## 10.42 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

### 10.42.1 Constructor & Destructor Documentation

#### 10.42.1.1 CAPICryptoFactory()

```
gdcM::CAPICryptoFactory::CAPICryptoFactory (
    CryptoLib id )
```

### 10.42.2 Member Function Documentation

#### 10.42.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcM::CAPICryptoFactory::CreateCMSProvider ( ) [virtual]
```

Implements [gdcM::CryptoFactory](#).

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

- [gdcMCAPICryptoFactory.h](#)

## 10.43 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)

## Additional Inherited Members

### 10.43.1 Constructor & Destructor Documentation

#### 10.43.1.1 CAPICryptographicMessageSyntax()

```
gdcM::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )
```

#### 10.43.1.2 ~CAPICryptographicMessageSyntax()

```
gdcM::CAPICryptographicMessageSyntax::~~CAPICryptographicMessageSyntax ( )
```

### 10.43.2 Member Function Documentation

#### 10.43.2.1 Decrypt()

```
bool gdcM::CAPICryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).



### 10.43.2.2 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](#).

### 10.43.2.3 GetCipherType()

```
CipherTypes gdcmm::CAPICryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.43.2.4 GetInitialized()

```
bool gdcmm::CAPICryptographicMessageSyntax::GetInitialized ( ) const [inline]
```

### 10.43.2.5 ParseCertificateFile()

```
bool gdcmm::CAPICryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.43.2.6 ParseKeyFile()

```
bool gdcmm::CAPICryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

#### 10.43.2.7 SetCipherType()

```
void gdcM::CAPICryptographicMessageSyntax::SetCipherType (
    CipherTypes type )
```

#### 10.43.2.8 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](#)

## 10.44 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`)

### Public Attributes

- [UIComp AffectedSOPClassUID](#)
- `uint16_t` [MessageID](#)

#### 10.44.1 Detailed Description

[CEchoRQ](#).

this file defines the messages for the cecho action

#### 10.44.2 Member Function Documentation

##### 10.44.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::CEchoRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [virtual]
```

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

### 10.44.3 Member Data Documentation

#### 10.44.3.1 AffectedSOPClassUID

[UIComp](#) `gdcm::network::CEchoRQ::AffectedSOPClassUID`

#### 10.44.3.2 MessageID

`uint16_t gdcm::network::CEchoRQ::MessageID`

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

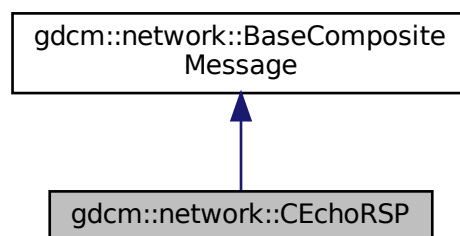
- [gdcmCEchoMessages.h](#)
- [gdcmDIMSE.h](#)

## 10.45 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)`

#### 10.45.1 Detailed Description

[CEchoRSP](#) this file defines the messages for the cecho action.

#### 10.45.2 Member Function Documentation

##### 10.45.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CEchoRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

- [gdcmCEchoMessages.h](#)

## 10.46 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

### 10.46.1 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](#)

## 10.47 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)

### 10.47.1 Detailed Description

[CFindCancelRQ](#) this file defines the messages for the cfind action.

### 10.47.2 Member Function Documentation

#### 10.47.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

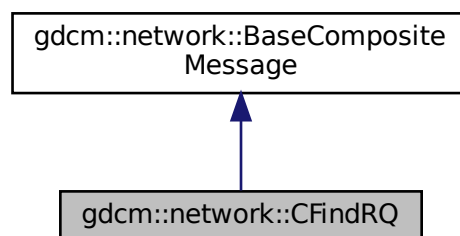
- [gdcmCFindMessages.h](#)

## 10.48 gdcm::network::CFindRQ Class Reference

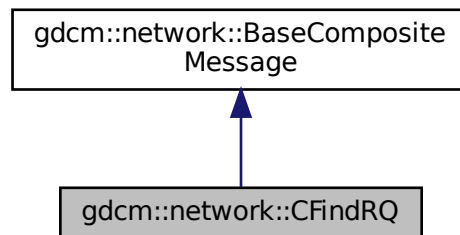
[CFindRQ](#).

```
#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`)

### 10.48.1 Detailed Description

[CFindRQ](#).

this file defines the messages for the cfind action

### 10.48.2 Member Function Documentation

#### 10.48.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::CFindRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [virtual]
```

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

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

- [gdcmCFindMessages.h](#)

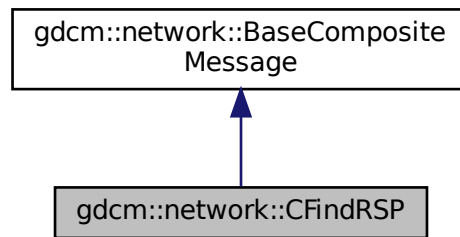


## 10.49 gdcm::network::CFindRSP Class Reference

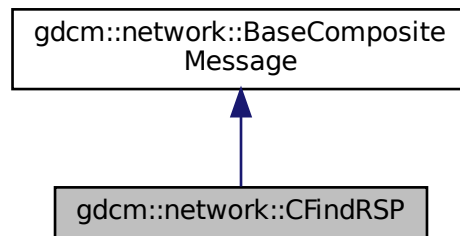
[CFindRSP](#) this file defines the messages for the cfind action.

```
#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)`

#### 10.49.1 Detailed Description

[CFindRSP](#) this file defines the messages for the cfind action.

## 10.49.2 Member Function Documentation

### 10.49.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcM::network::CFindRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

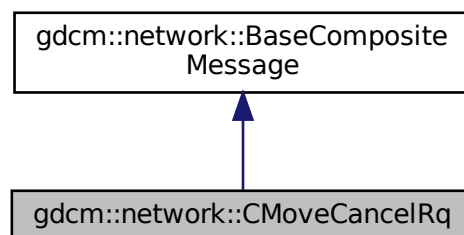
The documentation for this class was generated from the following file:

- [gdcMCFindMessages.h](#)

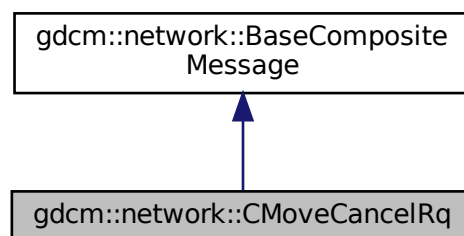
## 10.50 gdcM::network::CMoveCancelRq Class Reference

```
#include <gdcMCMoveMessages.h>
```

Inheritance diagram for gdcM::network::CMoveCancelRq:



Collaboration diagram for gdcM::network::CMoveCancelRq:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet` (const [DataSet](#) \*inDataSet)

### 10.50.1 Member Function Documentation

#### 10.50.1.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

- [gdcmCMoveMessages.h](#)

## 10.51 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`)

### 10.51.1 Detailed Description

[CMoveRQ](#).

this file defines the messages for the cmove action

### 10.51.2 Member Function Documentation

#### 10.51.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::CMoveRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [virtual]
```

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

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

- [gdcmCMoveMessages.h](#)

## 10.52 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)`

#### 10.52.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

## 10.52.2 Member Function Documentation

### 10.52.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

- [gdcmCMoveMessages.h](#)

## 10.53 gdcm::Codec Class Reference

[Codec](#) class.

```
#include <gdcmCodec.h>
```

Inheritance diagram for `gdcm::Codec`:



Collaboration diagram for gdcm::Codec:



## Additional Inherited Members

### 10.53.1 Detailed Description

[Codec](#) class.

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

- [gdcmCodec.h](#)

## 10.54 gdcm::Coder Class Reference

[Coder](#).

```
#include <gdcmCoder.h>
```

Inheritance diagram for gdcm::Coder:



## Public Member Functions

- virtual [~Coder](#) ()
- 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)

### 10.54.1 Detailed Description

[Coder](#).

### 10.54.2 Constructor & Destructor Documentation

#### 10.54.2.1 ~Coder()

```
virtual gdcm::Coder::~~Coder ( ) [inline], [virtual]
```

### 10.54.3 Member Function Documentation

#### 10.54.3.1 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::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGLSCCodec](#), [gdcm::ImageCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), [gdcm::RAWCodec](#), [gdcm::AudioCodec](#), and [gdcm::PDFCodec](#).



## 10.54.3.2 Code()

```
virtual bool gdcm::Coder::Code (
    DataElement const & in_,
    DataElement & out_ ) [inline], [virtual]
```

Code.

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::KAKADUCodec](#), and [gdcm::RAWCodec](#).

## 10.54.3.3 InternalCode()

```
virtual bool gdcm::Coder::InternalCode (
    const char * bv,
    unsigned long len,
    std::ostream & os ) [inline], [protected], [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

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

- [gdcmCoder.h](#)

## 10.55 gdcm::CodeString Class Reference

[CodeString](#).

```
#include <gdcmCodeString.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 [value\\_type](#) \*s)
- [CodeString](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- [CodeString](#) (const [InternalClass](#) &s, [size\\_type](#) pos=0, [size\\_type](#) n=[InternalClass](#)::npos)
- [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)

### 10.55.1 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.

### 10.55.2 Member Typedef Documentation

#### 10.55.2.1 const\_iterator

```
typedef InternalClass::const_iterator gdcm::CodeString::const_iterator
```

#### 10.55.2.2 const\_reference

```
typedef InternalClass::const_reference gdcm::CodeString::const_reference
```

#### 10.55.2.3 const\_reverse\_iterator

```
typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator
```

#### 10.55.2.4 difference\_type

```
typedef InternalClass::difference_type gdcm::CodeString::difference_type
```

#### 10.55.2.5 iterator

```
typedef InternalClass::iterator gdcm::CodeString::iterator
```

#### 10.55.2.6 pointer

```
typedef InternalClass::pointer gdcm::CodeString::pointer
```

#### 10.55.2.7 reference

```
typedef InternalClass::reference gdcm::CodeString::reference
```

#### 10.55.2.8 reverse\_iterator

```
typedef InternalClass::reverse\_iterator gdcm::CodeString::reverse_iterator
```

#### 10.55.2.9 size\_type

```
typedef InternalClass::size\_type gdcm::CodeString::size_type
```

#### 10.55.2.10 value\_type

```
typedef InternalClass::value\_type gdcm::CodeString::value_type
```

### 10.55.3 Constructor & Destructor Documentation

#### 10.55.3.1 CodeString() [1/4]

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

[CodeString](#) constructors.

#### 10.55.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
    const value\_type * s ) [inline]
```

#### 10.55.3.3 CodeString() [3/4]

```
gdcm::CodeString::CodeString (
    const value\_type * s,
    size\_type n ) [inline]
```

#### 10.55.3.4 CodeString() [ 4 / 4 ]

```
gdcm::CodeString::CodeString (
    const InternalClass & s,
    size\_type pos = 0,
    size\_type n = InternalClass::npos ) [inline]
```

### 10.55.4 Member Function Documentation

#### 10.55.4.1 GetAsString()

```
std::string gdcm::CodeString::GetAsString ( ) const [inline]
```

Return the full code string as std::string.

#### 10.55.4.2 IsValid()

```
bool gdcm::CodeString::IsValid ( ) const
```

Check if [CodeString](#) obj is correct..

#### 10.55.4.3 Size()

```
size\_type gdcm::CodeString::Size ( ) const [inline]
```

Return the size of the string.

#### 10.55.4.4 TrimInternal()

```
std::string gdcm::CodeString::TrimInternal ( ) const [inline], [protected]
```

### 10.55.5 Friends And Related Function Documentation

### 10.55.5.1 operator!=

```
bool operator!= (
    const CodeString & ref,
    const CodeString & cs ) [friend]
```

### 10.55.5.2 operator<<

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

### 10.55.5.3 operator==

```
bool operator== (
    const CodeString & ref,
    const CodeString & cs ) [friend]
```

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

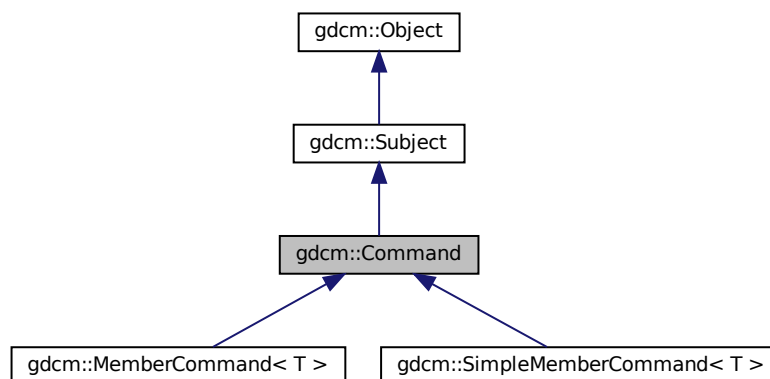
- [gdcmCodeString.h](#)

## 10.56 gdcm::Command Class Reference

[Command](#) superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcmm::Command:



### Public Member Functions

- virtual void [Execute](#) ([Subject](#) \*caller, const [Event](#) &event)=0  
*Abstract method that defines the action to be taken by the command.*
- virtual void [Execute](#) (const [Subject](#) \*caller, const [Event](#) &event)=0

### Protected Member Functions

- [Command](#) ()
- [~Command](#) ()

#### 10.56.1 Detailed Description

[Command](#) superclass for callback/observer methods.

See also

[Subject](#)

#### 10.56.2 Constructor & Destructor Documentation

### 10.56.2.1 Command()

```
gdcM::Command::Command ( ) [protected]
```

### 10.56.2.2 ~Command()

```
gdcM::Command::~~Command ( ) [protected]
```

## 10.56.3 Member Function Documentation

### 10.56.3.1 Execute() [1/2]

```
virtual void gdcM::Command::Execute (
    Subject * caller,
    const Event & event ) [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in [gdcM::SimpleMemberCommand< T >](#), and [gdcM::MemberCommand< T >](#).

### 10.56.3.2 Execute() [2/2]

```
virtual void gdcM::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 [gdcM::SimpleMemberCommand< T >](#), and [gdcM::MemberCommand< T >](#).

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

- [gdcMCommand.h](#)



## 10.57 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](#) ()
- [~CommandDataSet](#) ()
- 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.*

## Friends

- `std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)`

## Additional Inherited Members

### 10.57.1 Detailed Description

Class to represent a [Command DataSet](#).

See also

[DataSet](#)

### 10.57.2 Constructor & Destructor Documentation

#### 10.57.2.1 [CommandDataSet\(\)](#)

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

#### 10.57.2.2 [~CommandDataSet\(\)](#)

```
gdcm::CommandDataSet::~~CommandDataSet ( ) [inline]
```

References [gdcm::operator<<\(\)](#).

### 10.57.3 Member Function Documentation

#### 10.57.3.1 [Insert\(\)](#)

```
void gdcm::CommandDataSet::Insert (
    const DataElement & de ) [inline]
```

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

### 10.57.3.2 Read()

```
std::istream& gdcm::CommandDataSet::Read (
    std::istream & is )
```

Read.

### 10.57.3.3 Replace()

```
void gdcm::CommandDataSet::Replace (
    const DataElement & de ) [inline]
```

References [gdcm::DataElement::GetTag\(\)](#).

### 10.57.3.4 Write()

```
std::ostream& gdcm::CommandDataSet::Write (
    std::ostream & os ) const
```

Write.

## 10.57.4 Friends And Related Function Documentation

### 10.57.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CommandDataSet & _val ) [friend]
```

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

- [gdcmCommandDataSet.h](#)

## 10.58 gdcm::network::CompositeMessageFactory Class Reference

[CompositeMessageFactory](#).

```
#include <gdcmCompositeMessageFactory.h>
```

## Static Public Member Functions

- static std::vector< [PresentationDataValue](#) > [ConstructCEchoRQ](#) (const [ULConnection](#) &inConnection)
- static std::vector< [PresentationDataValue](#) > [ConstructCFindRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCMoveRQ](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \*inRootQuery)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRQ](#) (const [ULConnection](#) &inConnection, const [File](#) &file, bool writeDataSet=true)
- static std::vector< [PresentationDataValue](#) > [ConstructCStoreRSP](#) (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)

### 10.58.1 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).

### 10.58.2 Member Function Documentation

#### 10.58.2.1 ConstructCEchoRQ()

```
static std::vector<PresentationDataValue> gdcmm::network::CompositeMessageFactory::ConstructC↵
EchoRQ (
    const ULConnection & inConnection ) [static]
```

#### 10.58.2.2 ConstructCFindRQ()

```
static std::vector<PresentationDataValue> gdcmm::network::CompositeMessageFactory::ConstructC↵
FindRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

## 10.58.2.3 ConstructCMoveRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↵
MoveRQ (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

## 10.58.2.4 ConstructCStoreRQ()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↵
StoreRQ (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

## 10.58.2.5 ConstructCStoreRSP()

```
static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructC↵
StoreRSP (
    const DataSet * inDataSet,
    const BasePDU * inPC ) [static]
```

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

- [gdcmCompositeMessageFactory.h](#)

## 10.59 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=NULL, const char \*call=NULL)
- static bool [CFind](#) (const char \*remote, uint16\_t portno, const [BaseRootQuery](#) \*query, std::vector< [DataSet](#) > &retDataSets, const char \*aetitle=NULL, const char \*call=NULL)
- static bool [CMove](#) (const char \*remote, uint16\_t portno, const [BaseRootQuery](#) \*query, uint16\_t portscp, const char \*aetitle=NULL, const char \*call=NULL, const char \*outputdir=NULL)
- static [BaseRootQuery](#) \* [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [DataSet](#) &queryds, [EQueryType](#) queryType=eFind)
- static [BaseRootQuery](#) \* [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [KeyValue↔PairArrayType](#) &keys, [EQueryType](#) queryType=eFind)
- static bool [CStore](#) (const char \*remote, uint16\_t portno, const [Directory::FileNamesType](#) &filenames, const char \*aetitle=NULL, const char \*call=NULL)

### 10.59.1 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)

### 10.59.2 Member Typedef Documentation

#### 10.59.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

#### 10.59.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

### 10.59.3 Member Function Documentation

#### 10.59.3.1 CEcho()

```
static bool gdcmm::CompositeNetworkFunctions::CEcho (
    const char * remote,
    uint16_t portno,
    const char * aetitle = NULL,
    const char * call = NULL ) [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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	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.

#### 10.59.3.2 CFind()

```
static bool gdcmm::CompositeNetworkFunctions::CFind (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle = NULL,
    const char * call = NULL ) [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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	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.

**10.59.3.3 CMove()**

```
static bool gdcml::CompositeNetworkFunctions::CMove (
    const char * remote,
    uint16_t portno,
    const BaseRootQuery * query,
    uint16_t portscp,
    const char * aetitle = NULL,
    const char * call = NULL,
    const char * outputdir = NULL ) [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**

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
<i>outputdir</i>	is not set default to current dir ('.')

**Returns**

true if it worked.

**10.59.3.4 ConstructQuery() [1/2]**

```
static BaseRootQuery* gdcml::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

<i>inMove</i> ).	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)
------------------	--

## 10.59.3.5 ConstructQuery() [2/2]

```
static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (
    ERootType inRootType,
    EQueryLevel inQueryLevel,
    const KeyValuePairArrayType & keys,
    EQueryType queryType = eFind ) [static]
```

## Deprecated

## 10.59.3.6 CStore()

```
static bool gdcm::CompositeNetworkFunctions::CStore (
    const char * remote,
    uint16_t portno,
    const Directory::FileNamesType & filenames,
    const char * aetitle = NULL,
    const char * call = NULL ) [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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	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

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

- [gdcmCompositeNetworkFunctions.h](#)

## 10.60 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

**Public Member Functions**

- [ConstCharWrapper](#) (const char \*i=0)
- [operator const char \\* \(\) const](#)

### 10.60.1 Detailed Description

Do not use me.

### 10.60.2 Constructor & Destructor Documentation

#### 10.60.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (  
    const char * i = 0 ) [inline]
```

### 10.60.3 Member Function Documentation

#### 10.60.3.1 operator const char \*()

```
gdcm::ConstCharWrapper::operator const char * ( ) const [inline]
```

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

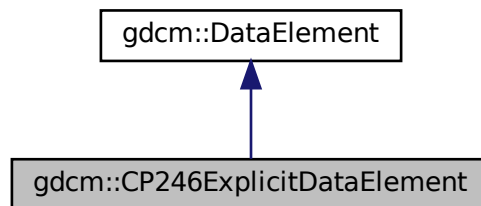
- [gdcmConstCharWrapper.h](#)

## 10.61 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)`

## Additional Inherited Members

### 10.61.1 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

### 10.61.2 Member Function Documentation

#### 10.61.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength ( ) const
```

#### 10.61.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::Read (
    std::istream & is )
```

#### 10.61.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue (
    std::istream & is )
```

## 10.61.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::CP246ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

## 10.61.2.5 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:

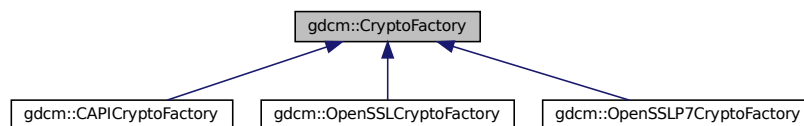
- [gdcmCP246ExplicitDataElement.h](#)

## 10.62 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,  
[OPENSSL7](#) = 3 }

## Public Member Functions

- virtual [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()=0

## Static Public Member Functions

- static [CryptoFactory](#) \* [GetFactoryInstance](#) ([CryptoLib](#) id=DEFAULT)

## Protected Member Functions

- [CryptoFactory](#) ([CryptoLib](#) id)
- [CryptoFactory](#) ()
- [~CryptoFactory](#) ()

### 10.62.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independant way the user specified crypto engine. It can be:

- CAPI (windows only)
- OPENSSL (portable)
- OPENSSLP7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSLP7 when older OpenSSL is used.

### 10.62.2 Member Enumeration Documentation

#### 10.62.2.1 [CryptoLib](#)

```
enum gdcm::CryptoFactory::CryptoLib
```

#### Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSLP7	

### 10.62.3 Constructor & Destructor Documentation

#### 10.62.3.1 CryptoFactory() [1/2]

```
gdcm::CryptoFactory::CryptoFactory (
    CryptoLib id ) [inline], [protected]
```

References [gdcmErrorMacro](#).

#### 10.62.3.2 CryptoFactory() [2/2]

```
gdcm::CryptoFactory::CryptoFactory ( ) [inline], [protected]
```

#### 10.62.3.3 ~CryptoFactory()

```
gdcm::CryptoFactory::~~CryptoFactory ( ) [inline], [protected]
```

### 10.62.4 Member Function Documentation

#### 10.62.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax* gdcm::CryptoFactory::CreateCMSProvider ( ) [pure virtual]
```

Implemented in [gdcm::OpenSSLCryptoFactory](#), [gdcm::OpenSSL7CryptoFactory](#), and [gdcm::CAPICryptoFactory](#).

#### 10.62.4.2 GetFactoryInstance()

```
static CryptoFactory* gdcm::CryptoFactory::GetFactoryInstance (
    CryptoLib id = DEFAULT ) [static]
```

Examples:

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

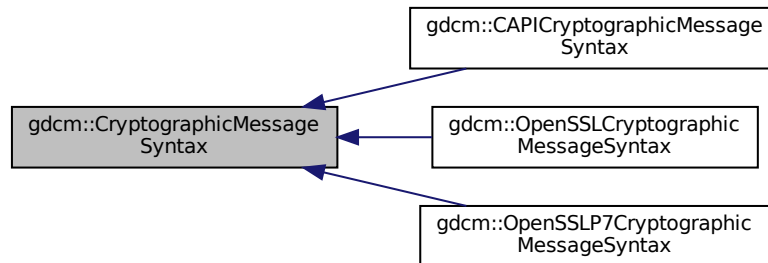
The documentation for this class was generated from the following file:

- [gdcmCryptoFactory.h](#)

## 10.63 gdcmm::CryptographicMessageSyntax Class Reference

```
#include <gdcmmCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcmm::CryptographicMessageSyntax:



### Public Types

- enum [CipherTypes](#) {  
[DES3\\_CIPHER](#),  
[AES128\\_CIPHER](#),  
[AES192\\_CIPHER](#),  
[AES256\\_CIPHER](#) }

### Public Member Functions

- [CryptographicMessageSyntax](#) ()
- virtual [~CryptographicMessageSyntax](#) ()
- 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
- 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

#### 10.63.1 Member Enumeration Documentation

##### 10.63.1.1 CipherTypes

```
enum gdcmm::CryptographicMessageSyntax::CipherTypes
```



## Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

## 10.63.2 Constructor & Destructor Documentation

### 10.63.2.1 CryptographicMessageSyntax()

```
gdcM::CryptographicMessageSyntax::CryptographicMessageSyntax ( ) [inline]
```

### 10.63.2.2 ~CryptographicMessageSyntax()

```
virtual gdcM::CryptographicMessageSyntax::~~CryptographicMessageSyntax ( ) [inline], [virtual]
```

## 10.63.3 Member Function Documentation

### 10.63.3.1 Decrypt()

```
virtual bool gdcM::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 [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::CAPICryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

### 10.63.3.2 Encrypt()

```
virtual bool gdcM::CryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [pure virtual]
```

create a CMS envelopedData structure

Implemented in [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::CAPICryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

### 10.63.3.3 GetCipherType()

```
virtual CipherTypes gdcM::CryptographicMessageSyntax::GetCipherType ( ) const [pure virtual]
```

Implemented in [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::CAPICryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

### 10.63.3.4 ParseCertificateFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::CAPICryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

### 10.63.3.5 ParseKeyFile()

```
virtual bool gdcM::CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [pure virtual]
```

Implemented in [gdcM::OpenSSLP7CryptographicMessageSyntax](#), [gdcM::CAPICryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

## 10.63.3.6 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [pure virtual]
```

Implemented in [gdcm::OpenSSLP7CryptographicMessageSyntax](#), and [gdcm::OpenSSLCryptographicMessageSyntax](#).

## 10.63.3.7 SetPassword()

```
virtual bool gdcm::CryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [pure virtual]
```

Implemented in [gdcm::OpenSSLP7CryptographicMessageSyntax](#), [gdcm::CAPICryptographicMessageSyntax](#), and [gdcm::OpenSSLCryptographicMessageSyntax](#).

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

- [gdcmCryptographicMessageSyntax.h](#)

## 10.64 gdcm::CSAElement Class Reference

Class to represent a CSA [Element](#).

```
#include <gdcmCSAElement.h>
```

Collaboration diagram for gdcm::CSAElement:



## Public Member Functions

- [CSAElement](#) (unsigned int kf=0)
- [CSAElement](#) (const [CSAElement](#) &\_val)
- const [ByteValue](#) \* [GetByteValue](#) () const
- unsigned int [GetKey](#) () const  
*Set/Get Key.*
- const char \* [GetName](#) () const  
*Set/Get Name.*
- unsigned int [GetNoOfItems](#) () const  
*Set/Get NoOfItems.*
- unsigned int [GetSyngoDT](#) () const  
*Set/Get SyngoDT.*
- [Value](#) const & [GetValue](#) () const  
*Set/Get Value (bytes array, SQ of items, SQ of fragments):*
- [Value](#) & [GetValue](#) ()
- 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)
- 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)`

### 10.64.1 Detailed Description

Class to represent a CSA [Element](#).

See also

[CSAHeader](#)

Examples:

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

### 10.64.2 Member Typedef Documentation

#### 10.64.2.1 DataPtr

```
typedef SmartPointer<Value> gdcmm::CSAElement::DataPtr [protected]
```

### 10.64.3 Constructor & Destructor Documentation

#### 10.64.3.1 CSAElement() [1/2]

```
gdcmm::CSAElement::CSAElement (
    unsigned int kf = 0 ) [inline]
```

References `gdcmm::operator<<()`.

#### 10.64.3.2 CSAElement() [2/2]

```
gdcmm::CSAElement::CSAElement (
    const CSAElement & _val ) [inline]
```

## 10.64.4 Member Function Documentation

### 10.64.4.1 GetByteValue()

```
const ByteValue* gdcM::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](#).

### 10.64.4.2 GetKey()

```
unsigned int gdcM::CSAElement::GetKey ( ) const [inline]
```

Set/Get Key.

Referenced by operator<().

### 10.64.4.3 GetName()

```
const char* gdcM::CSAElement::GetName ( ) const [inline]
```

Set/Get Name.

### 10.64.4.4 GetNoOfItems()

```
unsigned int gdcM::CSAElement::GetNoOfItems ( ) const [inline]
```

Set/Get NoOfItems.

#### 10.64.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT ( ) const [inline]
```

Set/Get SyngoDT.

#### 10.64.4.6 GetValue() [1/2]

```
Value const& gdcm::CSAElement::GetValue ( ) const [inline]
```

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples:

[csa2img.cxx](#).

#### 10.64.4.7 GetValue() [2/2]

```
Value& gdcm::CSAElement::GetValue ( ) [inline]
```

#### 10.64.4.8 GetVM()

```
const VM& gdcm::CSAElement::GetVM ( ) const [inline]
```

Set/Get [VM](#).

#### 10.64.4.9 GetVR()

```
VR const& gdcm::CSAElement::GetVR ( ) const [inline]
```

Set/Get [VR](#).

#### 10.64.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty ( ) const [inline]
```

Check if CSA [Element](#) is empty.

Examples:

[csa2img.cxx](#).

#### 10.64.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
    const CSAElement & de ) const [inline]
```

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

#### 10.64.4.12 operator=()

```
CSAElement& gdcm::CSAElement::operator= (
    const CSAElement & de ) [inline]
```

References [DataField](#), [KeyField](#), [NameField](#), [NoOfItemsField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

#### 10.64.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
    const CSAElement & de ) const [inline]
```

References [KeyField](#), [NameField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

#### 10.64.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
    const char * array,
    VL length ) [inline]
```

Set.



**10.64.4.15 SetKey()**

```
void gdcm::CSAElement::SetKey (
    unsigned int key ) [inline]
```

**10.64.4.16 SetName()**

```
void gdcm::CSAElement::SetName (
    const char * name ) [inline]
```

**10.64.4.17 SetNoOfItems()**

```
void gdcm::CSAElement::SetNoOfItems (
    unsigned int items ) [inline]
```

**10.64.4.18 SetSyngoDT()**

```
void gdcm::CSAElement::SetSyngoDT (
    unsigned int syngodt ) [inline]
```

**10.64.4.19 SetValue()**

```
void gdcm::CSAElement::SetValue (
    Value const & vl ) [inline]
```

**10.64.4.20 SetVM()**

```
void gdcm::CSAElement::SetVM (
    const VM & vm ) [inline]
```

#### 10.64.4.21 SetVR()

```
void gdcM::CSAElement::SetVR (
    VR const & vr ) [inline]
```

### 10.64.5 Friends And Related Function Documentation

#### 10.64.5.1 operator<<

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

### 10.64.6 Member Data Documentation

#### 10.64.6.1 DataField

```
DataPtr gdcM::CSAElement::DataField [protected]
```

Referenced by gdcM::operator<<(), and operator=().

#### 10.64.6.2 KeyField

```
unsigned int gdcM::CSAElement::KeyField [protected]
```

Referenced by gdcM::operator<<(), operator=(), and operator==().

#### 10.64.6.3 NameField

```
std::string gdcM::CSAElement::NameField [protected]
```

Referenced by gdcM::operator<<(), operator=(), and operator==().

#### 10.64.6.4 NoOfItemsField

```
unsigned int gdcm::CSAElement::NoOfItemsField [protected]
```

Referenced by `gdcm::operator<<()`, and `operator=()`.

#### 10.64.6.5 SyngoDTField

```
unsigned int gdcm::CSAElement::SyngoDTField [protected]
```

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

#### 10.64.6.6 ValueMultiplicityField

```
VM gdcm::CSAElement::ValueMultiplicityField [protected]
```

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

#### 10.64.6.7 VRField

```
VR gdcm::CSAElement::VRField [protected]
```

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

- [gdcmCSAElement.h](#)

## 10.65 gdcm::CSAHeader Class Reference

Class for [CSAHeader](#).

```
#include <gdcmCSAHeader.h>
```

## Public Types

- enum [CSAHeaderType](#) {  
[UNKNOWN](#) = 0,  
[SV10](#),  
[NOMAGIC](#),  
[DATASET\\_FORMAT](#),  
[INTERFILE](#),  
[ZEROED\\_OUT](#) }

*Divers format of [CSAHeader](#) as found 'in the wild'.*

## Public Member Functions

- [CSAHeader](#) ()
- [~CSAHeader](#) ()
- 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)*
- template<typename TSwap >  
std::istream & [Read](#) (std::istream &is)
- template<typename TSwap >  
const std::ostream & [Write](#) (std::ostream &os) const

## 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)

### 10.65.1 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](http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf)

#### Examples:

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

### 10.65.2 Member Enumeration Documentation

#### 10.65.2.1 CSAHeaderType

enum `gdcm::CSAHeader::CSAHeaderType`

Divers format of [CSAHeader](#) as found 'in the wild'.

#### Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

### 10.65.3 Constructor & Destructor Documentation

#### 10.65.3.1 CSAHeader()

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

#### 10.65.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~~CSAHeader ( ) [inline]
```

### 10.65.4 Member Function Documentation

#### 10.65.4.1 FindCSAELEMENTByName()

```
bool gdcm::CSAHeader::FindCSAELEMENTByName (
    const char * name )
```

Return true if the CSA element matching 'name' is found or not

#### Warning

Case Sensitive

#### Examples:

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

#### 10.65.4.2 GetCSADataInfo()

```
static const PrivateTag& gdcm::CSAHeader::GetCSADataInfo ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: [PrivateTag](#)(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

#### 10.65.4.3 GetCSAEnd()

```
const CSAElement& gdcm::CSAHeader::GetCSAEnd ( ) const [protected]
```

#### 10.65.4.4 GetCSAElementByName()

```
const CSAElement& gdcm::CSAHeader::GetCSAElementByName (
    const char * name )
```

Return the [CSAElement](#) corresponding to name 'name'

##### Warning

Case Sensitive

##### Examples:

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).

#### 10.65.4.5 GetCSAImageHeaderInfoTag()

```
static const PrivateTag& gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Image](#) Header This is: [PrivateTag](#)(0x0029,0x0010,"SIEMENS CSA HEADER");

##### Examples:

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [PublicDict.cxx](#).

#### 10.65.4.6 GetCSASeriesHeaderInfoTag()

```
static const PrivateTag& gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]
```

Return the private tag used by SIEMENS to store the CSA [Series](#) Header This is: [PrivateTag](#)(0x0029,0x0020,"SIEMENS CSA HEADER");

##### Examples:

[MrProtocol.cxx](#).

#### 10.65.4.7 GetDataSet()

```
const DataSet& gdcm::CSAHeader::GetDataSet ( ) const [inline]
```

Return the [DataSet](#) output (use only if Format == DATASET\_FORMAT )

#### 10.65.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const
```

return the format of the [CSAHeader](#) SV10 and NOMAGIC are equivalent.

#### 10.65.4.9 GetInterfile()

```
const char* gdcm::CSAHeader::GetInterfile ( ) const [inline]
```

Return the string output (use only if Format == Interfile)

#### 10.65.4.10 GetMrProtocol()

```
bool gdcm::CSAHeader::GetMrProtocol (
    const DataSet & ds,
    MrProtocol & mrProtocol )
```

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

**Examples:**

[MrProtocol.cxx](#).

#### 10.65.4.11 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
    DataElement const & de )
```

Decode the [CSAHeader](#) from element 'de'.

**Examples:**

[csa2img.cxx](#), [DumpSiemensBase64.cxx](#), and [MrProtocol.cxx](#).



#### 10.65.4.12 Print()

```
void gdcm::CSAHeader::Print (
    std::ostream & os ) const
```

Print the [CSAHeader](#) (use only if Format == SV10 or NOMAGIC)

Examples:

[csa2img.cxx](#).

Referenced by `gdcm::operator<<()`.

#### 10.65.4.13 Read()

```
template<typename TSwap >
std::istream& gdcm::CSAHeader::Read (
    std::istream & is )
```

#### 10.65.4.14 Write()

```
template<typename TSwap >
const std::ostream& gdcm::CSAHeader::Write (
    std::ostream & os ) const
```

### 10.65.5 Friends And Related Function Documentation

#### 10.65.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CSAHeader & d ) [friend]
```

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

- [gdcmCSAHeader.h](#)

## 10.66 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](#) ()
- void [AddCSAHeaderDictEntry](#) (const [CSAHeaderDictEntry](#) &de)
- [ConstIterator](#) [Begin](#) () const
- [ConstIterator](#) [End](#) () const
- const [CSAHeaderDictEntry](#) & [GetCSAHeaderDictEntry](#) (const char \*name) const
- bool [IsEmpty](#) () const

### Protected Member Functions

- void [LoadDefault](#) ()

### Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [CSAHeaderDict](#) &\_val)

#### 10.66.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

Examples:

[MrProtocol.cxx](#).

#### 10.66.2 Member Typedef Documentation

### 10.66.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator
```

### 10.66.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator
```

### 10.66.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry
```

## 10.66.3 Constructor & Destructor Documentation

### 10.66.3.1 CSAHeaderDict()

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

References `gdcm::operator<<()`.

## 10.66.4 Member Function Documentation

### 10.66.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (
    const CSAHeaderDictEntry & de ) [inline]
```

### 10.66.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin ( ) const [inline]
```

#### 10.66.4.3 End()

```
ConstIterator gdcM::CSAHeaderDict::End ( ) const [inline]
```

#### 10.66.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry& gdcM::CSAHeaderDict::GetCSAHeaderDictEntry (
    const char * name ) const [inline]
```

#### Examples:

[MrProtocol.cxx](#).

#### 10.66.4.5 IsEmpty()

```
bool gdcM::CSAHeaderDict::IsEmpty ( ) const [inline]
```

#### 10.66.4.6 LoadDefault()

```
void gdcM::CSAHeaderDict::LoadDefault ( ) [protected]
```

### 10.66.5 Friends And Related Function Documentation

#### 10.66.5.1 Dicts

```
friend class Dicts [friend]
```

#### 10.66.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const CSAHeaderDict & _val ) [friend]
```

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

- [gdcMCSAHeaderDict.h](#)

## 10.67 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](#) &entry) 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)

#### 10.67.1 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](#).

## 10.67.2 Constructor & Destructor Documentation

### 10.67.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
    const char * name = "",
    VR const & vr = VR::INVALID,
    VM const & vm = VM::VM0,
    const char * desc = "" ) [inline]
```

References `gdcm::operator<<()`.

## 10.67.3 Member Function Documentation

### 10.67.3.1 GetDescription()

```
const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]
```

Set/Get Description.

### 10.67.3.2 GetName()

```
const char* gdcm::CSAHeaderDictEntry::GetName ( ) const [inline]
```

Set/Get Name.

Referenced by `operator<()`.

### 10.67.3.3 GetVM()

```
const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]
```

Set/Get VM.

#### 10.67.3.4 GetVR()

```
const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const [inline]
```

Set/Get VR.

#### 10.67.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
    const CSAHeaderDictEntry & entry ) const [inline]
```

References GetName().

#### 10.67.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
    const char * desc ) [inline]
```

#### 10.67.3.7 SetName()

```
void gdcm::CSAHeaderDictEntry::SetName (
    const char * name ) [inline]
```

#### 10.67.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
    VM const & vm ) [inline]
```

#### 10.67.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
    const VR & vr ) [inline]
```

## 10.67.4 Friends And Related Function Documentation

### 10.67.4.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & _os,  
    const CSAHeaderDictEntry & _val ) [friend]
```

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

- [gdcmCSAHeaderDictEntry.h](#)

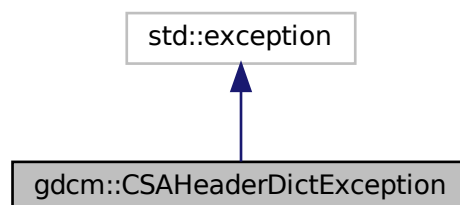
## 10.68 gdcm::CSAHeaderDictException Class Reference

```
#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](#)

## 10.69 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)

### 10.69.1 Detailed Description

[CStoreRQ](#).

this file defines the messages for the cecho action

### 10.69.2 Member Function Documentation

#### 10.69.2.1 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:

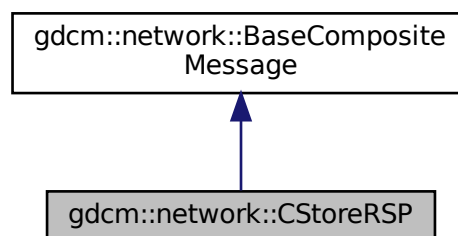
- [gdcM CStoreMessages.h](#)

## 10.70 gdcM::network::CStoreRSP Class Reference

[CStoreRSP](#) this file defines the messages for the cecho action.

```
#include <gdcM CStoreMessages.h>
```

Inheritance diagram for gdcM::network::CStoreRSP:



Collaboration diagram for gdcmm::network::CStoreRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)

#### 10.70.1 Detailed Description

[CStoreRSP](#) this file defines the messages for the cecho action.

#### 10.70.2 Member Function Documentation

##### 10.70.2.1 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](#)

## 10.71 gdcm::Curve Class Reference

[Curve](#) class to handle element 50xx,3000 [Curve](#) Data.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



### Public Member Functions

- [Curve](#) ()
- [Curve](#) ([Curve](#) const &ov)
- [~Curve](#) ()
- 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
- 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)

### Static Public Member Functions

- static unsigned int [GetNumberOfCurves](#) ([DataSet](#) const &ds)

### Additional Inherited Members

#### 10.71.1 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

#### 10.71.2 Constructor & Destructor Documentation

**10.71.2.1 Curve()** [1/2]

```
gdcM::Curve::Curve ( )
```

**10.71.2.2 ~Curve()**

```
gdcM::Curve::~~Curve ( )
```

**10.71.2.3 Curve()** [2/2]

```
gdcM::Curve::Curve (
    Curve const & ov )
```

**10.71.3 Member Function Documentation****10.71.3.1 Decode()**

```
void gdcM::Curve::Decode (
    std::istream & is,
    std::ostream & os )
```

**10.71.3.2 GetAsPoints()**

```
void gdcM::Curve::GetAsPoints (
    float * array ) const
```

**10.71.3.3 GetCurveDataDescriptor()**

```
std::vector<unsigned short> const& gdcM::Curve::GetCurveDataDescriptor ( ) const
```

#### 10.71.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
```

#### 10.71.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions ( ) const
```

#### 10.71.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup ( ) const
```

#### 10.71.3.7 GetNumberOfCurves()

```
static unsigned int gdcm::Curve::GetNumberOfCurves (
    DataSet const & ds ) [static]
```

#### 10.71.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
```

#### 10.71.3.9 GetTypeOfData()

```
const char* gdcm::Curve::GetTypeOfData ( ) const
```

#### 10.71.3.10 GetTypeOfDataDescription()

```
const char* gdcm::Curve::GetTypeOfDataDescription ( ) const
```

**10.71.3.11 IsEmpty()**

```
bool gdcm::Curve::IsEmpty ( ) const
```

**10.71.3.12 Print()**

```
void gdcm::Curve::Print (
    std::ostream & ) const [virtual]
```

Reimplemented from [gdcm::Object](#).

**10.71.3.13 SetCoordinateStartValue()**

```
void gdcm::Curve::SetCoordinateStartValue (
    unsigned short v )
```

**10.71.3.14 SetCoordinateStepValue()**

```
void gdcm::Curve::SetCoordinateStepValue (
    unsigned short v )
```

**10.71.3.15 SetCurve()**

```
void gdcm::Curve::SetCurve (
    const char * array,
    unsigned int length )
```

**10.71.3.16 SetCurveDataDescriptor()**

```
void gdcm::Curve::SetCurveDataDescriptor (
    const uint16_t * values,
    size_t num )
```



**10.71.3.17 SetCurveDescription()**

```
void gdcm::Curve::SetCurveDescription (
    const char * curvedescription )
```

**10.71.3.18 SetDataValueRepresentation()**

```
void gdcm::Curve::SetDataValueRepresentation (
    unsigned short datavaluerepresentation )
```

**10.71.3.19 SetDimensions()**

```
void gdcm::Curve::SetDimensions (
    unsigned short dimensions )
```

**10.71.3.20 SetGroup()**

```
void gdcm::Curve::SetGroup (
    unsigned short group )
```

**10.71.3.21 SetNumberOfPoints()**

```
void gdcm::Curve::SetNumberOfPoints (
    unsigned short numberofpoints )
```

**10.71.3.22 SetTypeOfData()**

```
void gdcm::Curve::SetTypeOfData (
    const char * typeofdata )
```

### 10.71.3.23 Update()

```
void gdcM::Curve::Update (
    const DataElement & de )
```

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

- [gdcMCurve.h](#)

## 10.72 gdcM::DataElement Class Reference

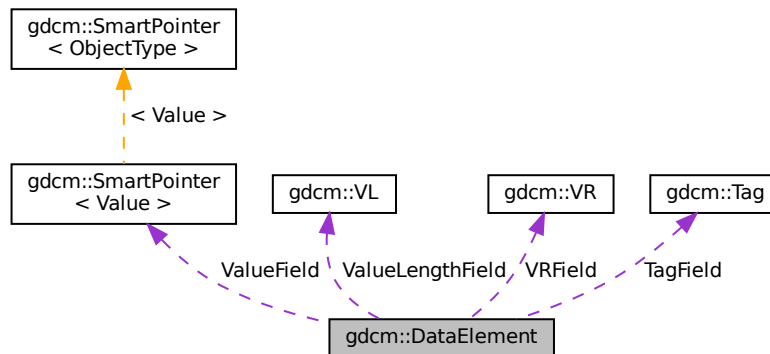
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#include <gdcMDataElement.h>
```

Inheritance diagram for gdcM::DataElement:



Collaboration diagram for gdcm::DataElement:



## Public Member Functions

- [DataElement](#) (const [Tag](#) &t=[Tag](#)(0), const [VL](#) &vl=0, const [VR](#) &vr=[VR::INVALID](#))
- [DataElement](#) (const [DataElement](#) &\_val)
- 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
- const [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) () const
- [SequenceOfFragments](#) \* [GetSequenceOfFragments](#) ()
- const [Tag](#) & [GetTag](#) () const
  - Get [Tag](#).*
- [Tag](#) & [GetTag](#) ()
- [Value](#) const & [GetValue](#) () const
  - Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):*
- [Value](#) & [GetValue](#) ()
- [SmartPointer](#)< [SequenceOfItems](#) > [GetValueAsSQ](#) () const
- const [VL](#) & [GetVL](#) () const
  - Get [VL](#).*
- [VL](#) & [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](#) &de)

- 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)

### 10.72.1 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:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImage↵  
HeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCD↵  
E.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [Extracting\\_All↵  
\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLong↵  
Seqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.↵  
cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicit↵  
LengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.72.2 Member Typedef Documentation

#### 10.72.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcm::DataElement::ValuePtr [protected]
```

### 10.72.3 Constructor & Destructor Documentation

### 10.72.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
    const Tag & t = Tag(0),
    const VL & vl = 0,
    const VR & vr = VR::INVALID ) [inline]
```

References gdcm::operator<<().

### 10.72.3.2 DataElement() [2/2]

```
gdcm::DataElement::DataElement (
    const DataElement & _val ) [inline]
```

## 10.72.4 Member Function Documentation

### 10.72.4.1 Clear()

```
void gdcm::DataElement::Clear ( ) [inline]
```

Clear Data Element (make Value empty and invalidate Tag & VR)

### 10.72.4.2 Empty()

```
void gdcm::DataElement::Empty ( ) [inline]
```

Make Data Element empty (no Value)

## 10.72.4.3 GetByteValue()

```
const ByteValue* gdcm::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](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDICOM.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::operator<<\(\)](#), [gdcm::Element< VR::OB, VM::VM1\\_n >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement\(\)](#), [gdcm::Element< TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).

## 10.72.4.4 GetLength()

```
template<typename TDE >
VL gdcm::DataElement::GetLength ( ) const [inline]
```

## 10.72.4.5 GetSequenceOfFragments() [1/2]

```
const SequenceOfFragments* gdcm::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:

[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

**10.72.4.6 GetSequenceOfFragments()** [2/2]

```
SequenceOfFragments* gdcM::DataElement::GetSequenceOfFragments ( )
```

**10.72.4.7 GetTag()** [1/2]

```
const Tag& gdcM::DataElement::GetTag ( ) const [inline]
```

Get [Tag](#).

Examples:

[DumpGEMSMovieGroup.cxx](#), [DuplicatePCDE.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcM::CommandDataSet::Insert\(\)](#), [gdcM::FileMetaInformation::Insert\(\)](#), [gdcM::DataSet::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\\_n >::SetFromDataElement\(\)](#).

**10.72.4.8 GetTag()** [2/2]

```
Tag& gdcM::DataElement::GetTag ( ) [inline]
```

**10.72.4.9 GetValue()** [1/2]

```
Value const& gdcM::DataElement::GetValue ( ) const [inline]
```

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

References [gdcMAssertAlwaysMacro](#).

Referenced by [gdcM::DataSet::InsertDataElement\(\)](#), [gdcM::Element< VR::OB, VM::VM1\\_n >::SetFromDataElement\(\)](#), and [gdcM::Element< TVR, VM::VM1\\_n >::SetFromDataElement\(\)](#).



## 10.72.4.10 GetValue() [2/2]

```
Value& gdcm::DataElement::GetValue ( ) [inline]
```

## 10.72.4.11 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](#), [ExtractEncryptedContent.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

## 10.72.4.12 GetVL() [1/2]

```
const VL& gdcm::DataElement::GetVL ( ) const [inline]
```

Get [VL](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfItems::Read()`, and `gdcm::SequenceOfFragments::ReadValue()`.

## 10.72.4.13 GetVL() [2/2]

```
VL& gdcm::DataElement::GetVL ( ) [inline]
```

#### 10.72.4.14 GetVR()

```
VR const& gdcm::DataElement::GetVR ( ) const [inline]
```

Get [VR](#) do not set [VR::SQ](#) on bytevalue data element

##### Examples:

[DuplicatePCDE.cxx](#), [GenFakeIdentifyFile.cxx](#), and [ReadAndDumpDICOMDIR2.cxx](#).

Referenced by `gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

#### 10.72.4.15 IsEmpty()

```
bool gdcm::DataElement::IsEmpty ( ) const [inline]
```

Check if Data [Element](#) is empty.

##### Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAIBugJPEGLS.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet()`.

#### 10.72.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

#### 10.72.4.17 operator<()

```
bool gdcm::DataElement::operator< (
    const DataElement & de ) const [inline]
```

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

#### 10.72.4.18 operator=()

```
DataElement& gdcm::DataElement::operator= (
    const DataElement & de ) [inline]
```

References [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

#### 10.72.4.19 operator==( )

```
bool gdcm::DataElement::operator== (
    const DataElement & de ) const [inline]
```

References [TagField](#), [ValueField](#), [ValueLengthField](#), and [VRField](#).

#### 10.72.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::Read (
    std::istream & is ) [inline]
```

Examples:

[DumpSiemensBase64.cxx](#).

#### 10.72.4.21 ReadOrSkip()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadOrSkip (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.22 ReadPreValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadPreValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.23 ReadValue()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValue (
    std::istream & is,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.24 ReadValueWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    std::set< Tag > const & skiptags ) [inline]
```

#### 10.72.4.25 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataElement::ReadWithLength (
    std::istream & is,
    VL & length ) [inline]
```

## 10.72.4.26 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:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.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](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by [gdcm::Element< VR::OB, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#).

## 10.72.4.27 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](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [GetSubSequenceData.cxx](#).

#### 10.72.4.28 SetValue()

```
void gdcm::DataElement::SetValue (
    Value const & vl ) [inline]
```

##### Warning

you need to set the ValueLengthField explicitly

##### Examples:

[DuplicatePCDE.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

References [gdcm::Value::GetLength\(\)](#).

#### 10.72.4.29 SetValueFieldLength()

```
void gdcm::DataElement::SetValueFieldLength (
    VL vl,
    bool readvalues ) [protected]
```

#### 10.72.4.30 SetVL()

```
void gdcm::DataElement::SetVL (
    const VL & vl ) [inline]
```

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

##### See also

[SetByteValue](#)

#### 10.72.4.31 SetVLToUndefined()

```
void gdcm::DataElement::SetVLToUndefined ( )
```

##### Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [NewSequence.cs](#).

## 10.72.4.32 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:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

References [gdcm::VR::IsVRFile\(\)](#).

Referenced by [gdcm::Element< VR::OB, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#), and [gdcm::Element< TVR, VM::VM1\\_n >::GetAsDataElement\(\)](#).

## 10.72.4.33 Write()

```
template<typename TDE , typename TSwap >
const std::ostream& gdcm::DataElement::Write (
    std::ostream & os ) const [inline]
```

## 10.72.5 Friends And Related Function Documentation

## 10.72.5.1 operator&lt;&lt;

```
std::ostream& operator<< (
    std::ostream & _os,
    const DataElement & _val ) [friend]
```

## 10.72.6 Member Data Documentation

#### 10.72.6.1 TagField

`Tag` `gdcm::DataElement::TagField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

#### 10.72.6.2 ValueField

`ValuePtr` `gdcm::DataElement::ValueField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

#### 10.72.6.3 ValueLengthField

`VL` `gdcm::DataElement::ValueLengthField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

#### 10.72.6.4 VRField

`VR` `gdcm::DataElement::VRField` [protected]

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

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

- [gdcmDataElement.h](#)



## 10.73 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](#)

## 10.74 gdcm::DataEvent Class Reference

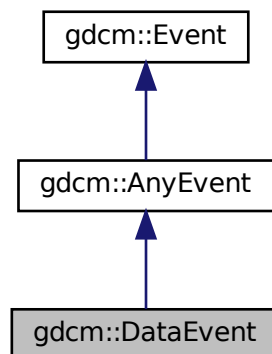
[DataEvent](#).

```
#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=0, size\_t len=0)
- [DataEvent](#) (const [Self](#) &s)
- virtual [~DataEvent](#) ()
- virtual bool [CheckEvent](#) (const ::gdcm::Event \*e) const
- const char \* [GetData](#) () const
- size\_t [GetDataLength](#) () const
- virtual const char \* [GetEventName](#) () const
- virtual ::gdcm::Event \* [MakeObject](#) () const
- void [SetData](#) (const char \*bytes, size\_t len)

### 10.74.1 Detailed Description

[DataEvent](#).

### 10.74.2 Member Typedef Documentation

#### 10.74.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

#### 10.74.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

### 10.74.3 Constructor & Destructor Documentation

#### 10.74.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (  
    const char * bytes = 0,  
    size_t len = 0 ) [inline]
```

### 10.74.3.2 ~DataEvent()

```
virtual gdcM::DataEvent::~~DataEvent ( ) [inline], [virtual]
```

### 10.74.3.3 DataEvent() [2/2]

```
gdcM::DataEvent::DataEvent (
    const Self & s ) [inline]
```

## 10.74.4 Member Function Documentation

### 10.74.4.1 CheckEvent()

```
virtual bool gdcM::DataEvent::CheckEvent (
    const ::gdcM::Event * e ) const [inline], [virtual]
```

### 10.74.4.2 GetData()

```
const char* gdcM::DataEvent::GetData ( ) const [inline]
```

### 10.74.4.3 GetDataLength()

```
size_t gdcM::DataEvent::GetDataLength ( ) const [inline]
```

### 10.74.4.4 GetEventName()

```
virtual const char* gdcM::DataEvent::GetEventName ( ) const [inline], [virtual]
```

Return the StringName associated with the event.

Implements [gdcM::Event](#).

## 10.74.4.5 MakeObject()

```
virtual ::gdcm::Event* gdcm::DataEvent::MakeObject ( ) const [inline], [virtual]
```

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

Implements [gdcm::Event](#).

## 10.74.4.6 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](#)

## 10.75 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 DataSet::const\_iterator [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef DataSet::iterator [Iterator](#)
- typedef DataSet::size\_type [SizeType](#)

## Public Member Functions

- [ConstIterator Begin](#) () const
- [Iterator Begin](#) ()
- void [Clear](#) ()
- template<typename TDE >  
unsigned int [ComputeGroupLength](#) ([Tag](#) const &tag) const
- [ConstIterator End](#) () const
- [Iterator End](#) ()
- 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 [Tag](#) &t) const
- const [DataElement](#) & [GetDataElement](#) (const [PrivateTag](#) &t) const  
*Return the dataelement.*
- const [DataElementSet](#) & [GetDES](#) () const
- [DataElementSet](#) & [GetDES](#) ()
- template<typename TDE >  
[VL GetLength](#) () const
- [MediaStorage GetMediaStorage](#) () const
- std::string [GetPrivateCreator](#) (const [Tag](#) &t) const  
*Return the private creator of the private tag 't':*
- 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 &val)
- 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)`

### 10.75.1 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](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixOrientation.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [VolumeSorter.cxx](#).

## 10.75.2 Member Typedef Documentation

### 10.75.2.1 ConstIterator

```
typedef DataSet::const_iterator gdcm::DataSet::ConstIterator
```

### 10.75.2.2 DataSet

```
typedef std::set<DataElement> gdcm::DataSet::DataSet
```

### 10.75.2.3 Iterator

```
typedef DataSet::iterator gdcm::DataSet::Iterator
```

### 10.75.2.4 SizeType

```
typedef DataSet::size_type gdcm::DataSet::SizeType
```

## 10.75.3 Member Function Documentation

### 10.75.3.1 Begin() [1/2]

```
ConstIterator gdcm::DataSet::Begin ( ) const [inline]
```

#### Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

Referenced by `gdcm::operator<<()`.



### 10.75.3.2 Begin() [2/2]

```
Iterator gdcm::DataSet::Begin ( ) [inline]
```

### 10.75.3.3 Clear()

```
void gdcm::DataSet::Clear ( ) [inline]
```

Referenced by `gdcm::Item::Read()`.

### 10.75.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
    const PrivateTag & t ) const [protected]
```

### 10.75.3.5 ComputeGroupLength()

```
template<typename TDE >
unsigned int gdcm::DataSet::ComputeGroupLength (
    Tag const & tag ) const [inline]
```

References `gdcm::Tag::GetElement()`, and `gdcm::Tag::GetGroup()`.

### 10.75.3.6 End() [1/2]

```
ConstIterator gdcm::DataSet::End ( ) const [inline]
```

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

### 10.75.3.7 End() [2/2]

```
Iterator gdcm::DataSet::End ( ) [inline]
```

**10.75.3.8 FindDataElement()** [1/2]

```
bool gdcM::DataSet::FindDataElement (
    const PrivateTag & t ) const
```

Look up if private tag 't' is present in the dataset:

**Examples:**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataSet()`, `gdcM::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`, and `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet()`.

**10.75.3.9 FindDataElement()** [2/2]

```
bool gdcM::DataSet::FindDataElement (
    const Tag & t ) const [inline]
```

**10.75.3.10 FindNextDataElement()**

```
const DataElement& gdcM::DataSet::FindNextDataElement (
    const Tag & t ) const [inline]
```

**Examples:**

[DuplicatePCDE.cxx](#).

## 10.75.3.11 GetDataElement() [1/2]

```
const DataElement& gdcm::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](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by [gdcm::Attribute< Group, Element, TVR, TVM >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::Set\(\)](#), [gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet\(\)](#), [gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet\(\)](#), and [gdcm::Attribute< Group, Element, TVR, VM::VM1\\_n >::SetFromDataSet\(\)](#).

## 10.75.3.12 GetDataElement() [2/2]

```
const DataElement& gdcm::DataSet::GetDataElement (
    const PrivateTag & t ) const
```

Return the dataelement.

## 10.75.3.13 GetDEEnd()

```
const DataElement& gdcm::DataSet::GetDEEnd ( ) const [protected]
```

## 10.75.3.14 GetDES() [1/2]

```
const DataElementSet& gdcm::DataSet::GetDES ( ) const [inline]
```

## Examples:

[ReadAndDumpDICOMDIR.cxx](#).

**10.75.3.15 GetDES()** [2/2]

```
DataElementSet& gdcm::DataSet::GetDES ( ) [inline]
```

**10.75.3.16 GetLength()**

```
template<typename TDE >
VL gdcm::DataSet::GetLength ( ) const [inline]
```

**10.75.3.17 GetMediaStorage()**

```
MediaStorage gdcm::DataSet::GetMediaStorage ( ) const
```

**10.75.3.18 GetPrivateCreator()**

```
std::string gdcm::DataSet::GetPrivateCreator (
    const Tag & t ) const
```

Return the private creator of the private tag 't':

**Examples:**

[DuplicatePCDE.cxx](#).

**10.75.3.19 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:**

[CreateJIPIDataset.cxx](#), [DumpSiemensBase64.cxx](#), [DuplicatePCDE.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [FakeImage\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [NewSequence.cs](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

References [gdcmErrorMacro](#), [gdcm::Tag::GetGroup\(\)](#), and [gdcm::DataElement::GetTag\(\)](#).

### 10.75.3.20 InsertDataElement()

```
void gdcm::DataSet::InsertDataElement (
    const DataElement & de ) [inline], [protected]
```

References [gdcmWarningMacro](#), [gdcm::Value::GetLength\(\)](#), [gdcm::DataElement::GetValue\(\)](#), [gdcm::DataElement::GetVL\(\)](#), [gdcm::DataElement::IsEmpty\(\)](#), and [gdcm::operator<<\(\)](#).

### 10.75.3.21 IsEmpty()

```
bool gdcm::DataSet::IsEmpty ( ) const [inline]
```

Returns if the dataset is empty.

Referenced by [gdcm::Item::Read\(\)](#).

### 10.75.3.22 operator()

```
const DataElement& gdcm::DataSet::operator() (
    uint16_t group,
    uint16_t element ) const [inline]
```

### 10.75.3.23 operator=()

```
DataSet& gdcm::DataSet::operator= (
    DataSet const & val ) [inline]
```

### 10.75.3.24 operator[]()

```
const DataElement& gdcm::DataSet::operator[] (
    const Tag & t ) const [inline]
```

### 10.75.3.25 Print()

```
void gdcmm::DataSet::Print (
    std::ostream & os,
    std::string const & indent = "" ) const [inline]
```

Referenced by `gdcmm::operator<<()`.

### 10.75.3.26 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcmm::DataSet::Read (
    std::istream & is )
```

### 10.75.3.27 ReadNested()

```
template<typename TDE , typename TSwap >
std::istream& gdcmm::DataSet::ReadNested (
    std::istream & is )
```

### 10.75.3.28 ReadSelectedPrivateTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcmm::DataSet::ReadSelectedPrivateTags (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    bool readvalues = true )
```

### 10.75.3.29 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcmm::DataSet::ReadSelectedPrivateTagsWithLength (
    std::istream & is,
    const std::set< PrivateTag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.75.3.30 ReadSelectedTags()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTags (
    std::istream & is,
    const std::set< Tag > & tags,
    bool readvalues = true )
```

### 10.75.3.31 ReadSelectedTagsWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (
    std::istream & is,
    const std::set< Tag > & tags,
    VL & length,
    bool readvalues = true )
```

### 10.75.3.32 ReadUpToTag()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTag (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags )
```

### 10.75.3.33 ReadUpToTagWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadUpToTagWithLength (
    std::istream & is,
    const Tag & t,
    std::set< Tag > const & skiptags,
    VL & length )
```

### 10.75.3.34 ReadWithLength()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::DataSet::ReadWithLength (
    std::istream & is,
    VL & length )
```

### 10.75.3.35 Remove()

```
SizeType gdcM::DataSet::Remove (
    const Tag & tag ) [inline]
```

Completely remove a dataelement from the dataset.

#### Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [pmsct\\_rgb1.cxx](#), [ReformatFile.cs](#), [rle2img.cxx](#), and [StandardizeFiles.cs](#).

### 10.75.3.36 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](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

References `gdcMAssertAlwaysMacro`.

### 10.75.3.37 ReplaceEmpty()

```
void gdcM::DataSet::ReplaceEmpty (
    const DataElement & de ) [inline]
```

Only replace a DICOM attribute when it is missing or empty.

References `gdcMAssertAlwaysMacro`.



### 10.75.3.38 Size()

```
SizeType gdcm::DataSet::Size ( ) const [inline]
```

#### Examples:

[DumpGEMSMovieGroup.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.75.3.39 Write()

```
template<typename TDE , typename TSwap >  
std::ostream const& gdcm::DataSet::Write (   
    std::ostream & os ) const
```

## 10.75.4 Friends And Related Function Documentation

### 10.75.4.1 CSAHeader

```
friend class CSAHeader [friend]
```

### 10.75.4.2 operator<<

```
std::ostream& operator<< (   
    std::ostream & _os,   
    const DataSet & val ) [friend]
```

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

- [gdcmDataSet.h](#)

## 10.76 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](#) ([DataSet](#) const \*ds=NULL)
- [DataSetEvent](#) (const [Self](#) &s)
- virtual [~DataSetEvent](#) ()
- virtual bool [CheckEvent](#) (const [::gdcm::Event](#) \*e) const
- [DataSet](#) const & [GetDataSet](#) () const
- virtual const char \* [GetEventName](#) () const
- virtual [::gdcm::Event](#) \* [MakeObject](#) () const

### 10.76.1 Detailed Description

[DataSetEvent](#).

Special type of event triggered during the [DataSet](#) store/move process

See also

### 10.76.2 Member Typedef Documentation

#### 10.76.2.1 Self

```
typedef DataSetEvent gdcm::DataSetEvent::Self
```

#### 10.76.2.2 Superclass

```
typedef AnyEvent gdcm::DataSetEvent::Superclass
```

### 10.76.3 Constructor & Destructor Documentation

#### 10.76.3.1 [DataSetEvent](#)() [1/2]

```
gdcm::DataSetEvent::DataSetEvent (  
    DataSet const * ds = NULL ) [inline]
```

### 10.76.3.2 ~DataSetEvent()

```
virtual gdcM::DataSetEvent::~~DataSetEvent ( ) [inline], [virtual]
```

### 10.76.3.3 DataSetEvent() [2/2]

```
gdcM::DataSetEvent::DataSetEvent (
    const Self & s ) [inline]
```

## 10.76.4 Member Function Documentation

### 10.76.4.1 CheckEvent()

```
virtual bool gdcM::DataSetEvent::CheckEvent (
    const ::gdcM::Event * e ) const [inline], [virtual]
```

### 10.76.4.2 GetDataSet()

```
DataSet const& gdcM::DataSetEvent::GetDataSet ( ) const [inline]
```

### 10.76.4.3 GetEventName()

```
virtual const char* gdcM::DataSetEvent::GetEventName ( ) const [inline], [virtual]
```

Return the StringName associated with the event.

Implements [gdcM::Event](#).

## 10.76.4.4 MakeObject()

```
virtual ::gdcm::Event* gdcm::DataSetEvent::MakeObject ( ) const [inline], [virtual]
```

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

Implements [gdcm::Event](#).

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

- [gdcmDataSetEvent.h](#)

## 10.77 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)

## 10.77.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level)

## 10.77.2 Member Function Documentation

## 10.77.2.1 ComputeVR()

```
static 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

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

- [gdcmDataSetHelper.h](#)

## 10.78 gdcm::Decoder Class Reference

[Decoder.](#)

```
#include <gdcmDecoder.h>
```

Inheritance diagram for gdcm::Decoder:



### Public Member Functions

- virtual [~Decoder](#) ()
- 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 &)

### 10.78.1 Detailed Description

[Decoder.](#)

### 10.78.2 Constructor & Destructor Documentation

### 10.78.2.1 ~Decoder()

```
virtual gdcm::Decoder::~~Decoder ( ) [inline], [virtual]
```

## 10.78.3 Member Function Documentation

### 10.78.3.1 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::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::ImageCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), [gdcm::AudioCodec](#), [gdcm::PDFCodec](#), [gdcm::PGXCodec](#), and [gdcm::KAKADUCodec](#).

### 10.78.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
    DataElement const & ,
    DataElement & ) [inline], [virtual]
```

Decode.

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::ImageCodec](#), [gdcm::DeltaEncodingCodec](#), [gdcm::KAKADUCodec](#), [gdcm::RAWCodec](#), [gdcm::AudioCodec](#), and [gdcm::PDFCodec](#).

### 10.78.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
    std::istream & ,
    std::ostream & ) [inline], [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::RLECodec](#), [gdcm::ImageCodec](#), [gdcm::RAWCodec](#), [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

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

- [gdcmDecoder.h](#)

## 10.79 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](#) ()

#### 10.79.1 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.

#### 10.79.2 Constructor & Destructor Documentation

##### 10.79.2.1 DefinedTerms()

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

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

- [gdcmDefinedTerms.h](#)

## 10.80 gdcm::Defs Class Reference

FIXME I do not like the name '[Defs](#)'.

```
#include <gdcmDefs.h>
```



## Public Member Functions

- [Defs](#) ()
- [~Defs](#) ()
- const [IOD](#) & [GetIODFromFile](#) (const [File](#) &file) const
- const [IODs](#) & [GetIODs](#) () const
- [IODs](#) & [GetIODs](#) ()
- const [Macros](#) & [GetMacros](#) () const
- [Macros](#) & [GetMacros](#) ()
- const [Modules](#) & [GetModules](#) () const
- [Modules](#) & [GetModules](#) ()
- [Type](#) [GetTypeFromTag](#) (const [File](#) &file, const [Tag](#) &tag) const
- bool [IsEmpty](#) () const
- bool [Verify](#) (const [File](#) &file) const
- bool [Verify](#) (const [DataSet](#) &ds) 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](#)

### 10.80.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

#### Note

bla

#### Examples:

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

### 10.80.2 Constructor & Destructor Documentation

#### 10.80.2.1 Defs()

```
gdcm::Defs::Defs ( )
```

#### 10.80.2.2 ~Defs()

```
gdcm::Defs::~~Defs ( )
```

### 10.80.3 Member Function Documentation

#### 10.80.3.1 GetIODFromFile()

```
const IOD& gdcm::Defs::GetIODFromFile (
    const File & file ) const
```

#### 10.80.3.2 GetIODNameFromMediaStorage()

```
static const char* gdcm::Defs::GetIODNameFromMediaStorage (
    MediaStorage const & ms ) [static]
```

#### Examples:

[GenerateStandardSOPClasses.cxx](#).

#### 10.80.3.3 GetIODs() [1/2]

```
const IODs& gdcm::Defs::GetIODs ( ) const [inline]
```

#### Examples:

[TraverseModules.cxx](#).

#### 10.80.3.4 GetIODs() [2/2]

```
IODs& gdcm::Defs::GetIODs ( ) [inline]
```

#### 10.80.3.5 GetMacros() [1/2]

```
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](#).

#### 10.80.3.6 GetMacros() [2/2]

```
Macros& gdcm::Defs::GetMacros ( ) [inline]
```

#### 10.80.3.7 GetModules() [1/2]

```
const Modules& gdcm::Defs::GetModules ( ) const [inline]
```

Examples:

[TraverseModules.cxx](#).

#### 10.80.3.8 GetModules() [2/2]

```
Modules& gdcm::Defs::GetModules ( ) [inline]
```

#### 10.80.3.9 GetTypeFromTag()

```
Type gdcM::Defs::GetTypeFromTag (
    const File & file,
    const Tag & tag ) const
```

#### 10.80.3.10 IsEmpty()

```
bool gdcM::Defs::IsEmpty ( ) const [inline]
```

#### 10.80.3.11 LoadDefaults()

```
void gdcM::Defs::LoadDefaults ( ) [protected]
```

#### 10.80.3.12 LoadFromFile()

```
void gdcM::Defs::LoadFromFile (
    const char * filename ) [protected]
```

#### 10.80.3.13 Verify() [1/2]

```
bool gdcM::Defs::Verify (
    const File & file ) const
```

#### 10.80.3.14 Verify() [2/2]

```
bool gdcM::Defs::Verify (
    const DataSet & ds ) const
```

### 10.80.4 Friends And Related Function Documentation

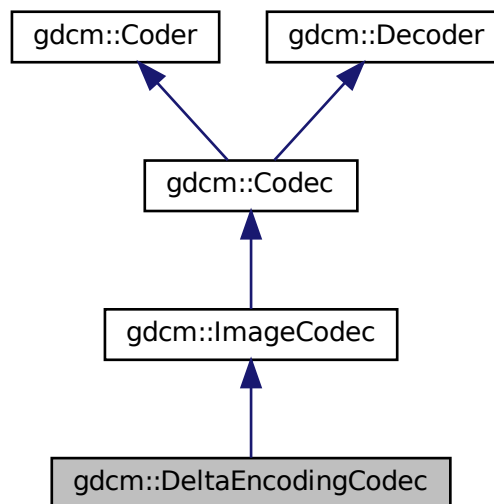
```
friend class Global [friend]
```

- [gdcmDefs.h](#)

**DeltaEncodingCodec** compression used by some private vendor.

```
#include <gdcMDeltaEncodingCodec.h>
```

Inheritance diagram for gdcm::DeltaEncodingCodec:



## Public Member Functions

- [DeltaEncodingCodec](#) ()
- [~DeltaEncodingCodec](#) ()
- bool [CanDecode](#) ([TransferSyntax](#) const &ts)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

## Protected Member Functions

- bool [Decode](#) (std::istream &is, std::ostream &os)

## Additional Inherited Members

### 10.81.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

### 10.81.2 Constructor & Destructor Documentation

#### 10.81.2.1 [DeltaEncodingCodec](#)()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )
```

#### 10.81.2.2 [~DeltaEncodingCodec](#)()

```
gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ( )
```

### 10.81.3 Member Function Documentation

#### 10.81.3.1 [CanDecode](#)()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
    TransferSyntax const & ts )
```

### 10.81.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

### 10.81.3.3 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](#)

## 10.82 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

### Public Member Functions

- [DICOMDIR](#) ()
- [DICOMDIR](#) (const [FileSet](#) &fs)

### 10.82.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

### 10.82.2 Constructor & Destructor Documentation

### 10.82.2.1 DICOMDIR() [1/2]

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

### 10.82.2.2 DICOMDIR() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (
    const FileSet & fs ) [inline]
```

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

- [gdcmDICOMDIR.h](#)

## 10.83 gdcm::DICOMDIRGenerator Class Reference

[DICOMDIRGenerator](#) class.

```
#include <gdcmDICOMDIRGenerator.h>
```

### Public Types

- typedef [Directory::FilenameType](#) [FilenameType](#)
- 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](#) ([FilenameType](#) 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](#) ()

### 10.83.1 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

### 10.83.2 Member Typedef Documentation

#### 10.83.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

### 10.83.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

## 10.83.3 Constructor & Destructor Documentation

### 10.83.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )
```

### 10.83.3.2 ~DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ( )
```

## 10.83.4 Member Function Documentation

### 10.83.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]
```

### 10.83.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]
```

### 10.83.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]
```

#### 10.83.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]
```

#### 10.83.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ( )
```

Main function to generate the [DICOMDIR](#).

#### 10.83.4.6 GetFile()

```
File& gdcm::DICOMDIRGenerator::GetFile ( )
```

#### 10.83.4.7 GetScanner()

```
Scanner& gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]
```

#### 10.83.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
    const char * d )
```

Set the [File](#) Set ID.

##### Warning

this need to be a valid [VR::CS](#) value

#### 10.83.4.9 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.

#### 10.83.4.10 SetFilenames()

```
void gdcmm::DICOMDIRGenerator::SetFilenames (
    FilenamesType const & fns )
```

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

#### 10.83.4.11 SetRootDirectory()

```
void gdcmm::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:

- [gdcmmDICOMDIRGenerator.h](#)

## 10.84 gdcmm::Dict Class Reference

Class to represent a map of [DictEntry](#).

```
#include <gdcmmDict.h>
```

### Public Types

- typedef MapDictEntry::const\_iterator [ConstIterator](#)
- typedef MapDictEntry::iterator [Iterator](#)
- typedef std::map< [Tag](#), [DictEntry](#) > [MapDictEntry](#)

### Public Member Functions

- [Dict](#) ()
- 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

## Protected Member Functions

- void [LoadDefault](#) ()

## Friends

- class [Dicts](#)
- std::ostream & [operator<<](#) (std::ostream &\_os, const [Dict](#) &\_val)

### 10.84.1 Detailed Description

Class to represent a map of [DictEntry](#).

#### Note

bla TODO FIXME: For [Element](#) == 0x0 need to return Name = Group Length ValueRepresentation = UL Value↔ Multiplicity = 1

#### Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

### 10.84.2 Member Typedef Documentation

#### 10.84.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

#### 10.84.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

#### 10.84.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

### 10.84.3 Constructor & Destructor Documentation

#### 10.84.3.1 Dict()

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

References `gdcm::operator<<()`.

### 10.84.4 Member Function Documentation

#### 10.84.4.1 AddDictEntry()

```
void gdcm::Dict::AddDictEntry (
    const Tag & tag,
    const DictEntry & de ) [inline]
```

#### 10.84.4.2 Begin()

```
ConstIterator gdcm::Dict::Begin ( ) const [inline]
```

Examples:

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

#### 10.84.4.3 End()

```
ConstIterator gdcm::Dict::End ( ) const [inline]
```

Examples:

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

#### 10.84.4.4 GetDictEntry()

```
const DictEntry& gdcm::Dict::GetDictEntry (
    const Tag & tag ) const [inline]
```

Examples:

[GenFakelIdentifyFile.cxx](#), and [PublicDict.cxx](#).

#### 10.84.4.5 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](#).

#### 10.84.4.6 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](#).

#### 10.84.4.7 GetKeywordFromTag()

```
const char* gdcm::Dict::GetKeywordFromTag (
    Tag const & tag ) const [inline]
```

Function to return the Keyword from a [Tag](#).

#### 10.84.4.8 IsEmpty()

```
bool gdcM::Dict::IsEmpty ( ) const [inline]
```

Referenced by `gdcM::Dicts::IsEmpty()`.

#### 10.84.4.9 LoadDefault()

```
void gdcM::Dict::LoadDefault ( ) [protected]
```

### 10.84.5 Friends And Related Function Documentation

#### 10.84.5.1 Dicts

```
friend class Dicts [friend]
```

#### 10.84.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Dict & _val ) [friend]
```

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

- [gdcMDict.h](#)

## 10.85 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](#) ()

### 10.85.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embeded dict into shared lib (DICT\_DEFAULT)
- Debug mode (DICT\_DEBUG)
- XML dict (DICT\_XML)

Note

### 10.85.2 Member Enumeration Documentation

#### 10.85.2.1 OutputTypes

```
enum gdcm::DictConverter::OutputTypes
```

**Enumerator**

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

**10.85.3 Constructor & Destructor Documentation****10.85.3.1 DictConverter()**

```
gdcm::DictConverter::DictConverter ( )
```

**10.85.3.2 ~DictConverter()**

```
gdcm::DictConverter::~~DictConverter ( )
```

**10.85.4 Member Function Documentation****10.85.4.1 AddGroupLength()**

```
void gdcm::DictConverter::AddGroupLength ( ) [protected]
```

**10.85.4.2 Convert()**

```
void gdcm::DictConverter::Convert ( )
```

**10.85.4.3 ConvertToCXX()**

```
bool gdcm::DictConverter::ConvertToCXX (
    const char * raw,
    std::string & cxx ) [protected]
```

#### 10.85.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (
    const char * raw,
    std::string & cxx ) [protected]
```

#### 10.85.4.5 GetDictName()

```
const std::string& gdcm::DictConverter::GetDictName ( ) const
```

#### 10.85.4.6 GetInputFilename()

```
const std::string& gdcm::DictConverter::GetInputFilename ( ) const
```

#### 10.85.4.7 GetOutputFilename()

```
const std::string& gdcm::DictConverter::GetOutputFilename ( ) const
```

#### 10.85.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType ( ) const [inline]
```

#### 10.85.4.9 Readuint16()

```
static bool gdcm::DictConverter::Readuint16 (
    const char * raw,
    uint16_t & ov ) [static]
```

**10.85.4.10 ReadVM()**

```
static bool gdcm::DictConverter::ReadVM (
    const char * raw,
    VM::VMType & type ) [static]
```

**10.85.4.11 ReadVR()**

```
static bool gdcm::DictConverter::ReadVR (
    const char * raw,
    VR::VRType & type ) [static]
```

**10.85.4.12 SetDictName()**

```
void gdcm::DictConverter::SetDictName (
    const char * name )
```

**10.85.4.13 SetInputFileName()**

```
void gdcm::DictConverter::SetInputFileName (
    const char * filename )
```

**10.85.4.14 SetOutputFileName()**

```
void gdcm::DictConverter::SetOutputFileName (
    const char * filename )
```

**10.85.4.15 SetOutputType()**

```
void gdcm::DictConverter::SetOutputType (
    int type ) [inline]
```

## 10.85.4.16 WriteFooter()

```
void gdcm::DictConverter::WriteFooter ( ) [protected]
```

## 10.85.4.17 WriteHeader()

```
void gdcm::DictConverter::WriteHeader ( ) [protected]
```

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

- [gdcmDictConverter.h](#)

## 10.86 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)`

### 10.86.1 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](#), [GenFakeldentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

### 10.86.2 Constructor & Destructor Documentation

#### 10.86.2.1 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::operator<<\(\)](#).

### 10.86.3 Member Function Documentation

### 10.86.3.1 GetKeyword()

```
const char* gdcm::DictEntry::GetKeyword ( ) const [inline]
```

same as GetName but without spaces...

### 10.86.3.2 GetName()

```
const char* gdcm::DictEntry::GetName ( ) const [inline]
```

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

### 10.86.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired ( ) const [inline]
```

Set/Get Retired flag.

Examples:

[GenAllVR.cxx](#).

### 10.86.3.4 GetVM()

```
const VM& gdcm::DictEntry::GetVM ( ) const [inline]
```

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

#### 10.86.3.5 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()`.

#### 10.86.3.6 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)

#### 10.86.3.7 SetElementXX()

```
void gdcM::DictEntry::SetElementXX (
    bool v ) [inline]
```

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

#### 10.86.3.8 SetGroupXX()

```
void gdcM::DictEntry::SetGroupXX (
    bool v ) [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically)

#### 10.86.3.9 SetKeyword()

```
void gdcM::DictEntry::SetKeyword (
    const char * keyword ) [inline]
```



#### 10.86.3.10 SetName()

```
void gdcm::DictEntry::SetName (
    const char * name ) [inline]
```

#### 10.86.3.11 SetRetired()

```
void gdcm::DictEntry::SetRetired (
    bool retired ) [inline]
```

#### 10.86.3.12 SetVM()

```
void gdcm::DictEntry::SetVM (
    VM const & vm ) [inline]
```

#### 10.86.3.13 SetVR()

```
void gdcm::DictEntry::SetVR (
    const VR & vr ) [inline]
```

Referenced by `gdcm::PrivateDict::AddDictEntry()`.

### 10.86.4 Friends And Related Function Documentation

#### 10.86.4.1 Dict

```
friend class Dict [friend]
```

#### 10.86.4.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const DictEntry & _val ) [friend]
```

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

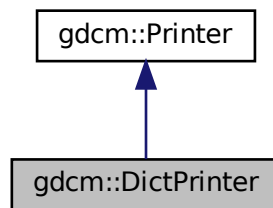
- [gdcmDictEntry.h](#)

## 10.87 gdcmmDictPrinter Class Reference

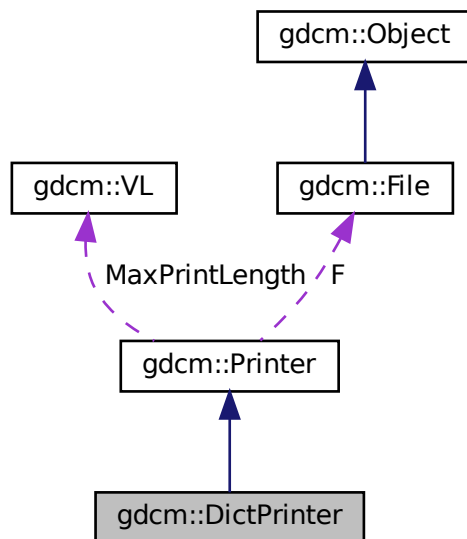
[DictPrinter](#) class.

```
#include <gdcmmDictPrinter.h>
```

Inheritance diagram for gdcmmDictPrinter:



Collaboration diagram for gdcmmDictPrinter:



## Public Member Functions

- [DictPrinter](#) ()
- [~DictPrinter](#) ()
- void [Print](#) (std::ostream &os)

## Protected Member Functions

- void [PrintDataElement2](#) (std::ostream &os, const [DataSet](#) &ds, const [DataElement](#) &ide)
- void [PrintDataSet2](#) (std::ostream &os, const [DataSet](#) &ds)

## Additional Inherited Members

### 10.87.1 Detailed Description

[DictPrinter](#) class.

### 10.87.2 Constructor & Destructor Documentation

#### 10.87.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ( )
```

#### 10.87.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~~DictPrinter ( )
```

### 10.87.3 Member Function Documentation

#### 10.87.3.1 Print()

```
void gdcm::DictPrinter::Print (
    std::ostream & os )
```

### 10.87.3.2 PrintDataElement2()

```
void gdcmm::DictPrinter::PrintDataElement2 (
    std::ostream & os,
    const DataSet & ds,
    const DataElement & ide ) [protected]
```

### 10.87.3.3 PrintDataSet2()

```
void gdcmm::DictPrinter::PrintDataSet2 (
    std::ostream & os,
    const DataSet & ds ) [protected]
```

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

- [gdcmmDictPrinter.h](#)

## 10.88 gdcmm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmmDicts.h>
```

### Public Member Functions

- [Dicts](#) ()
- [~Dicts](#) ()
- const [CSAHeaderDict](#) & [GetCSAHeaderDict](#) () const
- const [DictEntry](#) & [GetDictEntry](#) (const [Tag](#) &tag, const char \*owner=NULL) const  
*NOT THREAD SAFE.*
- const [DictEntry](#) & [GetDictEntry](#) (const [PrivateTag](#) &tag) const
- const [PrivateDict](#) & [GetPrivateDict](#) () const
- [PrivateDict](#) & [GetPrivateDict](#) ()
- const [Dict](#) & [GetPublicDict](#) () const
- bool [IsEmpty](#) () const

### 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)

### 10.88.1 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](#).

### 10.88.2 Member Enumeration Documentation

#### 10.88.2.1 ConstructorType

```
enum gdcmm::Dicts::ConstructorType [protected]
```

#### Enumerator

PHILIPS	
GEMS	
SIEMENS	

### 10.88.3 Constructor & Destructor Documentation

### 10.88.3.1 Dicts()

```
gdcmm::Dicts::Dicts ( )
```

### 10.88.3.2 ~Dicts()

```
gdcmm::Dicts::~~Dicts ( )
```

## 10.88.4 Member Function Documentation

### 10.88.4.1 GetConstructorString()

```
static const char* gdcmm::Dicts::GetConstructorString (
    ConstructorType type ) [static], [protected]
```

### 10.88.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict& gdcmm::Dicts::GetCSAHeaderDict ( ) const
```

#### Examples:

[MrProtocol.cxx](#).

### 10.88.4.3 GetDictEntry() [1/2]

```
const DictEntry& gdcmm::Dicts::GetDictEntry (
    const Tag & tag,
    const char * owner = NULL ) const
```

NOT 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](#).

#### 10.88.4.4 GetDictEntry() [2/2]

```
const DictEntry& gdcmm::Dicts::GetDictEntry (
    const PrivateTag & tag ) const
```

#### 10.88.4.5 GetPrivateDict() [1/2]

```
const PrivateDict& gdcmm::Dicts::GetPrivateDict ( ) const
```

#### 10.88.4.6 GetPrivateDict() [2/2]

```
PrivateDict& gdcmm::Dicts::GetPrivateDict ( )
```

#### 10.88.4.7 GetPublicDict()

```
const Dict& gdcmm::Dicts::GetPublicDict ( ) const
```

#### Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

#### 10.88.4.8 IsEmpty()

```
bool gdcmm::Dicts::IsEmpty ( ) const [inline]
```

References [gdcmm::Dict::IsEmpty\(\)](#).

#### 10.88.4.9 LoadDefaults()

```
void gdcmm::Dicts::LoadDefaults ( ) [protected]
```

## 10.88.5 Friends And Related Function Documentation

### 10.88.5.1 Global

```
friend class Global [friend]
```

### 10.88.5.2 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Dicts & d ) [friend]
```

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

- [gdcmDicts.h](#)

## 10.89 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 }



### 10.89.1 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)

### 10.89.2 Member Enumeration Documentation

#### 10.89.2.1 CommandTypes

```
enum gdcmm::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:

- [gdcmmDIMSE.h](#)

## 10.90 gdcM::DirectionCosines Class Reference

class to handle [DirectionCosines](#)

```
#include <gdcMDirectionCosines.h>
```

### Public Member Functions

- [DirectionCosines](#) ()
- [DirectionCosines](#) (const double dircos[6])
- [~DirectionCosines](#) ()
- 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 void [Normalize](#) (double v[3])  
*Normalize in-place.*

### 10.90.1 Detailed Description

class to handle [DirectionCosines](#)

Examples:

[DiscriminateVolume.cxx](#).

## 10.90.2 Constructor & Destructor Documentation

### 10.90.2.1 DirectionCosines() [1/2]

```
gdcmm::DirectionCosines::DirectionCosines ( )
```

### 10.90.2.2 DirectionCosines() [2/2]

```
gdcmm::DirectionCosines::DirectionCosines (
    const double dircos[6] )
```

### 10.90.2.3 ~DirectionCosines()

```
gdcmm::DirectionCosines::~~DirectionCosines ( )
```

## 10.90.3 Member Function Documentation

### 10.90.3.1 ComputeDistAlongNormal()

```
double gdcmm::DirectionCosines::ComputeDistAlongNormal (
    const double ipp[3] ) const
```

Compute the distance along the normal.

### 10.90.3.2 Cross()

```
void gdcmm::DirectionCosines::Cross (
    double z[3] ) const
```

Compute Cross product.

### 10.90.3.3 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](#).

### 10.90.3.4 Dot() [1/2]

```
double gdcm::DirectionCosines::Dot ( ) const
```

Compute Dot.

### 10.90.3.5 Dot() [2/2]

```
static double gdcm::DirectionCosines::Dot (
    const double x[3],
    const double y[3] ) [static]
```

Compute Dot.

### 10.90.3.6 IsValid()

```
bool gdcm::DirectionCosines::IsValid ( ) const
```

Return whether or not this is a valid direction cosines.

### 10.90.3.7 Normalize() [1/2]

```
void gdcm::DirectionCosines::Normalize ( )
```

Normalize in-place.

### 10.90.3.8 Normalize() [2/2]

```
static void gdcm::DirectionCosines::Normalize (
    double v[3] ) [static]
```

Normalize in-place.

### 10.90.3.9 operator const double \*()

```
gdcm::DirectionCosines::operator const double * ( ) const [inline]
```

Make the class behave like a const double \*.

### 10.90.3.10 Print()

```
void gdcm::DirectionCosines::Print (
    std::ostream & ) const
```

Print.

### 10.90.3.11 SetFromString()

```
bool gdcm::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](#)

## 10.91 gdcm::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](#) ()
- [~Directory](#) ()
- [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)

### 10.91.1 Detailed Description

Class for manipulation directories.

#### Note

This implementation provide a cross platform implementation for manipulating directores: 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:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8Qt↵Dir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcm.cxx](#), and [VolumeSorter.cxx](#).

## 10.91.2 Member Typedef Documentation

### 10.91.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

#### Examples:

[DiscriminateVolume.cxx](#).

### 10.91.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

## 10.91.3 Constructor & Destructor Documentation

### 10.91.3.1 Directory()

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

### 10.91.3.2 ~Directory()

```
gdcm::Directory::~~Directory ( ) [inline]
```

## 10.91.4 Member Function Documentation

#### 10.91.4.1 Explore()

```
unsigned int gdcM::Directory::Explore (
    FilenameType const & name,
    bool recursive ) [protected]
```

Return number of file found when 'recursive'ly exploring directory name

#### 10.91.4.2 GetDirectories()

```
FilenameType const& gdcM::Directory::GetDirectories ( ) const [inline]
```

Return the Directories traversed.

#### 10.91.4.3 GetFileNames()

```
FilenameType const& gdcM::Directory::GetFileNames ( ) const [inline]
```

Set/Get the file names within the directory.

#### Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcMOrthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcM.cxx](#), and [VolumeSorter.cxx](#).

#### 10.91.4.4 GetToplevel()

```
FilenameType const& gdcM::Directory::GetToplevel ( ) const [inline]
```

Get the name of the toplevel directory.



#### 10.91.4.5 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:

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

#### 10.91.4.6 Print()

```
void gdcmm::Directory::Print (
    std::ostream & os = std::cout ) const
```

Print.

#### Examples:

[SortImage.cxx](#).

Referenced by `gdcmm::operator<<()`.

### 10.91.5 Friends And Related Function Documentation

#### 10.91.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Directory & d ) [friend]
```

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

- [gdcmmDirectory.h](#)

## 10.92 gdcm::DirectoryHelper Class Reference

[DirectoryHelper](#).

```
#include <gdcmDirectoryHelper.h>
```

### Static Public Member Functions

- static [Directory::FilenameType](#) [GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType](#) [GetFileNamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FilenameType](#) [GetMRImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType](#) [GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType](#) [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)

### 10.92.1 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

### 10.92.2 Member Function Documentation

#### 10.92.2.1 GetCTImageSeriesUIDs()

```
static Directory::FilenameType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

### 10.92.2.2 GetFileNamesFromSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetFileNamesFromSeriesUIDs (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

Examples:

[GenerateRTSTRUCT.cxx](#).

### 10.92.2.3 GetFrameOfReference()

```
static std::string gdcm::DirectoryHelper::GetFrameOfReference (
    const std::vector< DataSet > & inDS ) [static]
```

### 10.92.2.4 GetMRImageSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
    const std::string & inDirectory ) [static]
```

### 10.92.2.5 GetRTStructSeriesUIDs()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
    const std::string & inDirectory ) [static]
```

Examples:

[GenerateRTSTRUCT.cxx](#).

### 10.92.2.6 GetSeriesUIDsBySOPClassUID()

```
static Directory::FileNamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
    const std::string & inDirectory,
    const std::string & inSOPClassUID ) [static]
```

#### 10.92.2.7 GetSOPClassUID()

```
static std::string gdcm::DirectoryHelper::GetSOPClassUID (
    const std::vector< DataSet > & inDS ) [static]
```

#### 10.92.2.8 GetStringValueFromTag()

```
static std::string gdcm::DirectoryHelper::GetStringValueFromTag (
    const Tag & t,
    const DataSet & ds ) [static]
```

#### 10.92.2.9 LoadImageFromFiles()

```
static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles (
    const std::string & inDirectory,
    const std::string & inSeriesUID ) [static]
```

#### 10.92.2.10 RetrieveSOPInstanceUIDFromIndex()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
    int inIndex,
    const std::vector< DataSet > & inDS ) [static]
```

#### 10.92.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
    double inZPos,
    const std::vector< DataSet > & inDS ) [static]
```

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

- [gdcmDirectoryHelper.h](#)

## 10.93 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

## Static Public Member Functions

- static const char \* [Generate](#) (const char \*input)

### 10.93.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

### 10.93.2 Member Function Documentation

#### 10.93.2.1 Generate()

```
static 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](#)

## 10.94 gdcm::Dumper Class Reference

[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for `gdcm::Dumper`:



### Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()

### Additional Inherited Members

#### 10.94.1 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

#### 10.94.2 Constructor & Destructor Documentation

## 10.94.2.1 Dumper()

```
gdcmm::Dumper::Dumper ( ) [inline]
```

## 10.94.2.2 ~Dumper()

```
gdcmm::Dumper::~~Dumper ( ) [inline]
```

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

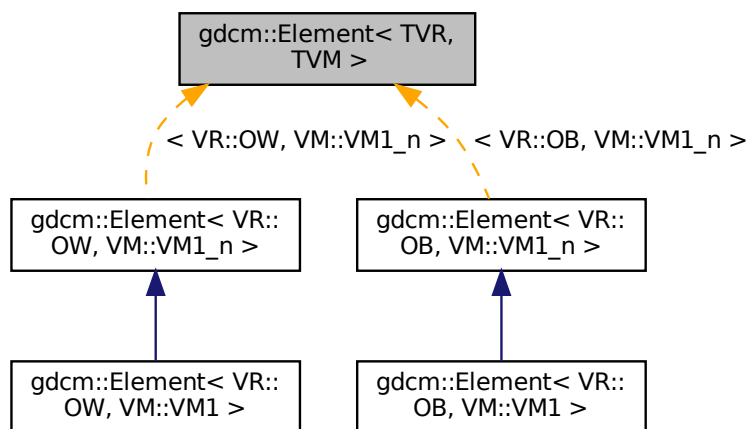
- [gdcmmDumper.h](#)

## 10.95 gdcmm::Element&lt; TVR, TVM &gt; Class Template Reference

[Element](#) class.

```
#include <gdcmmElement.h>
```

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



Collaboration diagram for `gdcm::Element< TVR, TVM >`:



## Public Types

- typedef `VRTToType< TVR >::Type` `Type`

## Public Member Functions

- `DataElement GetAsDataElement ()` const
- unsigned long `GetLength ()` const
- const `VRTToType< TVR >::Type & GetValue` (unsigned int idx=0) const
- `VRTToType< TVR >::Type & GetValue` (unsigned int idx=0)
- const `VRTToType< TVR >::Type * GetValues ()` const
- `VRTToType< 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 `VRTToType< 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)

### 10.95.1 Detailed Description

```
template<int TVR, int TVM>
class gdcm::Element< TVR, TVM >
```

[Element](#) class.

#### Note

TODO

#### Examples:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

### 10.95.2 Member Typedef Documentation

#### 10.95.2.1 Type

```
template<int TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

### 10.95.3 Member Function Documentation

#### 10.95.3.1 GetAsDataElement()

```
template<int TVR, int TVM>
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const [inline]
```

### 10.95.3.2 GetLength()

```
template<int TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]
```

### 10.95.3.3 GetValue() [1/2]

```
template<int TVR, int TVM>
const VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

### 10.95.3.4 GetValue() [2/2]

```
template<int TVR, int TVM>
VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (
    unsigned int idx = 0 ) [inline]
```

### 10.95.3.5 GetValues()

```
template<int TVR, int TVM>
const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]
```

### 10.95.3.6 GetVM()

```
template<int TVR, int TVM>
static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]
```

### 10.95.3.7 GetVR()

```
template<int TVR, int TVM>
static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]
```

### 10.95.3.8 operator[]()

```
template<int TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[] (
    unsigned int idx ) const [inline]
```

### 10.95.3.9 Print()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
    std::ostream & _os ) const [inline]
```

### 10.95.3.10 Read()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
    std::istream & _is ) [inline]
```

### 10.95.3.11 Set()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
    Value const & v ) [inline]
```

### 10.95.3.12 SetFromDataElement()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetFromDataElement (
    DataElement< TVR, TVM > const & de ) [inline]
```

### 10.95.3.13 SetNoSwap()

```
template<int TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

### 10.95.3.14 SetValue()

```
template<int TVR, int TVM>
void gdcmm::Element< TVR, TVM >::SetValue (
    typename VRToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

### 10.95.3.15 Write()

```
template<int TVR, int TVM>
void gdcmm::Element< TVR, TVM >::Write (
    std::ostream & _os ) const [inline]
```

## 10.95.4 Member Data Documentation

### 10.95.4.1 Internal

```
template<int TVR, int TVM>
VRToType<TVR>::Type gdcmm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by `gdcmm::Element< TVR, VM::VM1_n >::operator=()`.

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

- [gdcmmElement.h](#)

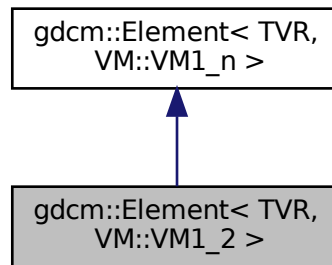
## 10.96 gdcmm::Element< TVR, VM::VM1\_2 > Class Template Reference

```
#include <gdcmmElement.h>
```

Inheritance diagram for `gdcmm::Element< TVR, VM::VM1_2 >`:



Collaboration diagram for gdcmm::Element< TVR, VM::VM1\_2 >:



## Public Types

- typedef [Element](#)< TVR, VM::VM1\_n > [Parent](#)

## Public Member Functions

- void [SetLength](#) (int len)

## Additional Inherited Members

### 10.96.1 Member Typedef Documentation

#### 10.96.1.1 Parent

```
template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM1_2 >::Parent
```

### 10.96.2 Member Function Documentation

### 10.96.2.1 SetLength()

```
template<int TVR>
void gdcM::Element< TVR, VM::VM1_2 >::SetLength (
    int len ) [inline]
```

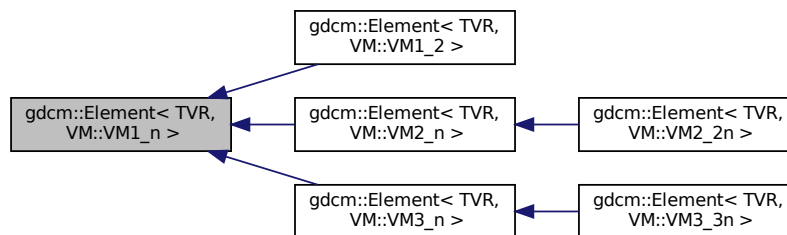
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

## 10.97 gdcM::Element< TVR, VM::VM1\_n > Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance 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
- const [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0) const
- [VRToType< TVR >::Type](#) & [GetValue](#) (unsigned int idx=0)
- [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)

## 10.97.1 Member Typedef Documentation

### 10.97.1.1 Type

```
template<int TVR>
typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1\_n >::Type
```

## 10.97.2 Constructor & Destructor Documentation

### 10.97.2.1 Element() [1/2]

```
template<int TVR>
gdcm::Element< TVR, VM::VM1\_n >::Element ( ) [inline], [explicit]
```

### 10.97.2.2 ~Element()

```
template<int TVR>
gdcm::Element< TVR, VM::VM1\_n >::~~Element ( ) [inline]
```

### 10.97.2.3 Element() [2/2]

```
template<int TVR>
gdcm::Element< TVR, VM::VM1\_n >::Element (
    const Element< TVR, VM::VM1\_n > &_val ) [inline]
```

### 10.97.3 Member Function Documentation

#### 10.97.3.1 GetAsDataElement()

```
template<int TVR>
DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]
```

References `gdcm::DataElement::GetVR()`, `gdcm::DataElement::SetByteValue()`, and `gdcm::DataElement::SetVR()`.

#### 10.97.3.2 GetLength()

```
template<int TVR>
unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]
```

#### 10.97.3.3 GetValue() [1/2]

```
template<int TVR>
const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) const [inline]
```

#### 10.97.3.4 GetValue() [2/2]

```
template<int TVR>
VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (
    unsigned int idx = 0 ) [inline]
```

#### 10.97.3.5 GetVM()

```
template<int TVR>
static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]
```



### 10.97.3.6 GetVR()

```
template<int TVR>
static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]
```

### 10.97.3.7 operator=()

```
template<int TVR>
Element& gdcm::Element< TVR, VM::VM1_n >::operator= (
    const Element< TVR, VM::VM1_n > & _val ) [inline]
```

References gdcm::Element< TVR, TVM >::Internal.

### 10.97.3.8 operator[]()

```
template<int TVR>
VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
    unsigned int idx ) const [inline]
```

### 10.97.3.9 Print()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
    std::ostream & _os ) const [inline]
```

### 10.97.3.10 Read()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
    std::istream & _is ) [inline]
```

**10.97.3.11 Set()**

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
    Value const & v ) [inline]
```

References `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::VRBINARY`.

**10.97.3.12 SetArray()**

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
    const Type * array,
    unsigned long len,
    bool save = false ) [inline]
```

**10.97.3.13 SetFromDataElement()**

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (
    DataElement< TVR, VM::VM1_n > const & de ) [inline]
```

References `gdcm::DataElement::GetByteValue()`, `gdcm::DataElement::GetValue()`, and `gdcm::DataElement::GetVR()`.

**10.97.3.14 SetLength()**

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
    unsigned long len ) [inline]
```

**10.97.3.15 SetNoSwap()**

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
    Value const & v ) [inline], [protected]
```

References `gdcm::ByteValue::GetLength()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::VRBINARY`.

## 10.97.3.16 SetValue()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
    typename VRTToType< TVR >::Type v,
    unsigned int idx = 0 ) [inline]
```

## 10.97.3.17 Write()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::Write (
    std::ostream & _os ) const [inline]
```

## 10.97.3.18 WriteASCII()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (
    std::ostream & os ) const [inline]
```

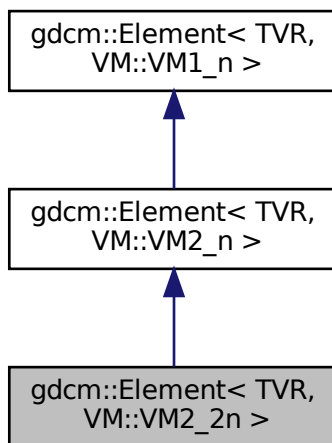
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

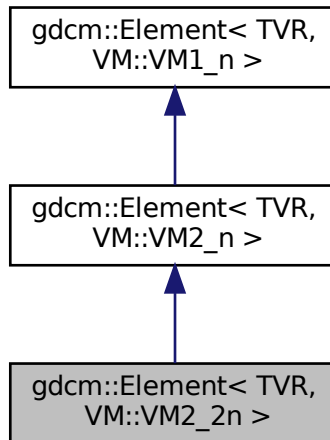
## 10.98 gdcm::Element&lt; TVR, VM::VM2\_2n &gt; Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2\_2n >:



Collaboration diagram for `gdcM::Element< TVR, VM::VM2_2n >`:



## Public Types

- typedef `Element< TVR, VM::VM2_n >` `Parent`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.98.1 Member Typedef Documentation

#### 10.98.1.1 Parent

```

template<int TVR>
typedef Element<TVR, VM::VM2_n> gdcM::Element< TVR, VM::VM2_2n >::Parent

```

### 10.98.2 Member Function Documentation

## 10.98.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM2_2n >::SetLength (
    int len ) [inline]
```

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

- [gdcmElement.h](#)

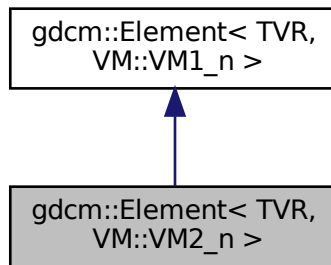
## 10.99 gdcm::Element&lt; TVR, VM::VM2\_n &gt; Class Template Reference

```
#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`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.99.1 Member Typedef Documentation

#### 10.99.1.1 Parent

```

template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcM::Element< TVR, VM::VM2_n >::Parent

```

### 10.99.2 Member Function Documentation

## 10.99.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM2_n >::SetLength (
    int len ) [inline]
```

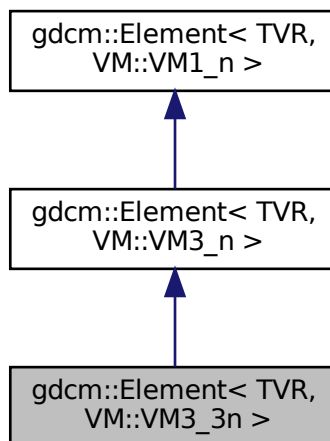
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

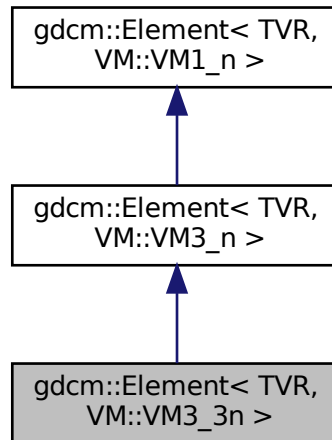
## 10.100 gdcm::Element&lt; TVR, VM::VM3\_3n &gt; Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3\_3n >:



Collaboration diagram for `gdcM::Element< TVR, VM::VM3_3n >`:



## Public Types

- typedef `Element< TVR, VM::VM3_n >` `Parent`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.100.1 Member Typedef Documentation

#### 10.100.1.1 Parent

```

template<int TVR>
typedef Element<TVR, VM::VM3_n> gdcM::Element< TVR, VM::VM3_3n >::Parent

```

### 10.100.2 Member Function Documentation



## 10.100.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
    int len ) [inline]
```

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

- [gdcmElement.h](#)

## 10.101 gdcm::Element&lt; TVR, VM::VM3\_n &gt; Class Template Reference

```
#include <gdcmElement.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`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 10.101.1 Member Typedef Documentation

#### 10.101.1.1 Parent

```

template<int TVR>
typedef Element<TVR, VM::VM1_n> gdcM::Element< TVR, VM::VM3_n >::Parent

```

### 10.101.2 Member Function Documentation

### 10.101.2.1 SetLength()

```
template<int TVR>
void gdcm::Element< TVR, VM::VM3_n >::SetLength (
    int len ) [inline]
```

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

- [gdcmElement.h](#)

## 10.102 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- unsigned long [GetLength](#) () const
- void [Print](#) (std::ostream &\_os) const

### Public Attributes

- char [Internal](#) [[VMToLength](#)< VM::VM5 >::Length \*sizeof([VRToType](#)< VR::AS >::Type)]

### 10.102.1 Member Function Documentation

#### 10.102.1.1 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]
```

#### 10.102.1.2 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
    std::ostream &_os ) const [inline]
```

### 10.102.2 Member Data Documentation

## 10.102.2.1 Internal

```
char gdcM::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType<
VR::AS >::Type)]
```

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

- [gdcMElement.h](#)

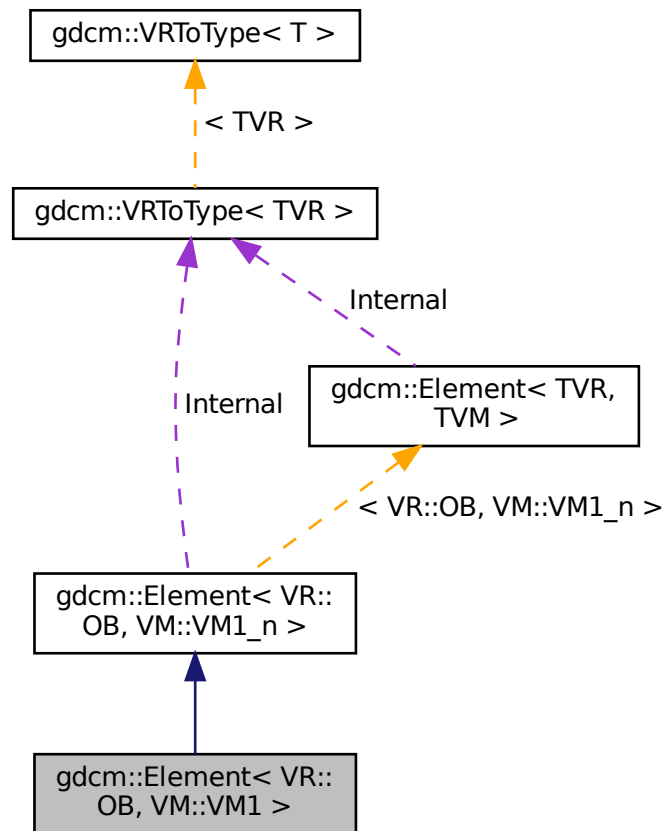
## 10.103 gdcM::Element&lt; VR::OB, VM::VM1 &gt; Class Template Reference

```
#include <gdcMElement.h>
```

Inheritance diagram for gdcM::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



### Additional Inherited Members

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

- [gdcmElement.h](#)

## 10.104 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< VR::OW, VM::VM1 >`:



Collaboration diagram for gdcm::Element< VR::OW, VM::VM1 >:



### Additional Inherited Members

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

- [gdcmElement.h](#)

## 10.105 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>
```

### 10.105.1 Detailed Description

```
template<int 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](#)

### 10.106 gdcm::ElementDisableCombinations< VR::OB, VM::VM1\_n > Class Template Reference

```
#include <gdcmElement.h>
```

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

- [gdcmElement.h](#)

### 10.107 gdcm::ElementDisableCombinations< VR::OW, VM::VM1\_n > Class Template Reference

```
#include <gdcmElement.h>
```

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

- [gdcmElement.h](#)

### 10.108 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)

### 10.108.1 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 Multiframe↔GrayscaleByteSecondaryCaptureImageStorage (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](#).

## 10.108.2 Member Enumeration Documentation

### 10.108.2.1 SOPClassUIDMode

enum [gdcm::EmptyMaskGenerator::SOPClassUIDMode](#)

#### Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

## 10.108.3 Constructor & Destructor Documentation

### 10.108.3.1 EmptyMaskGenerator()

[gdcm::EmptyMaskGenerator::EmptyMaskGenerator](#) ( )

### 10.108.3.2 ~EmptyMaskGenerator()

[gdcm::EmptyMaskGenerator::~EmptyMaskGenerator](#) ( )

## 10.108.4 Member Function Documentation

### 10.108.4.1 Execute()

bool [gdcm::EmptyMaskGenerator::Execute](#) ( )

Main loop.

Examples:

[EmptyMask.cxx](#).

#### 10.108.4.2 SetInputDirectory()

```
void gdcm::EmptyMaskGenerator::SetInputDirectory (
    const char * dirname )
```

Specify input directory.

Examples:

[EmptyMask.cxx](#).

#### 10.108.4.3 SetOutputDirectory()

```
void gdcm::EmptyMaskGenerator::SetOutputDirectory (
    const char * dirname )
```

Specify output directory.

Examples:

[EmptyMask.cxx](#).

#### 10.108.4.4 SetSOPClassUIDMode()

```
void gdcm::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:

- [gdcmEmptyMaskGenerator.h](#)

## 10.109 gdcm::EncapsulatedDocument Class Reference

[EncapsulatedDocument](#).

```
#include <gdcmEncapsulatedDocument.h>
```

## Public Member Functions

- [EncapsulatedDocument](#) ()

### 10.109.1 Detailed Description

[EncapsulatedDocument](#).

### 10.109.2 Constructor & Destructor Documentation

#### 10.109.2.1 EncapsulatedDocument()

```
gdcM::EncapsulatedDocument::EncapsulatedDocument ( ) [inline]
```

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

- [gdcMEncapsulatedDocument.h](#)

## 10.110 gdcM::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

```
#include <gdcMElement.h>
```

### 10.110.1 Detailed Description

```
template<int T>
class gdcM::EncodingImplementation< T >
```

[EncodingImplementation](#).

Note

TODO

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

- [gdcMElement.h](#)

## 10.111 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- template<>  
void [Write](#) (const float \*data, unsigned long length, std::ostream &\_os)
- 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)

### 10.111.1 Member Function Documentation

#### 10.111.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

#### 10.111.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

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

**10.111.1.3 ReadNoSwap()**

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

**10.111.1.4 Write()** [1/3]

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const T * data,
    unsigned long length,
    std::ostream & _os ) [inline], [static]
```

**10.111.1.5 Write()** [2/3]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const float * data,
    unsigned long length,
    std::ostream & _os ) [inline]
```

References `gdcm::to_string()`.

**10.111.1.6 Write()** [3/3]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
    const double * data,
    unsigned long length,
    std::ostream & _os ) [inline]
```

References `gdcm::to_string()`.

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

- [gdcmElement.h](#)

## 10.112 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference

```
#include <gdcmElement.h>
```

### 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)

### 10.112.1 Member Function Documentation

#### 10.112.1.1 Read()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

#### 10.112.1.2 ReadComputeLength()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
    T * data,
    unsigned int & length,
    std::istream & _is ) [inline], [static]
```

#### 10.112.1.3 ReadNoSwap()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
    T * data,
    unsigned long length,
    std::istream & _is ) [inline], [static]
```

#### 10.112.1.4 Write()

```
template<typename T >
static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
    const T * data,
    unsigned long length,
    std::ostream & _os ) [inline], [static]
```

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

- [gdcmElement.h](#)

### 10.113 gdcm::EndEvent Class Reference

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

Inheritance diagram for gdcm::EndEvent:





Collaboration diagram for gdcM::EndEvent:



### Additional Inherited Members

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

- [gdcMEvent.h](#)

## 10.114 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](#) ()

#### 10.114.1 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**.

## 10.114.2 Constructor & Destructor Documentation

### 10.114.2.1 EnumeratedValues()

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

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

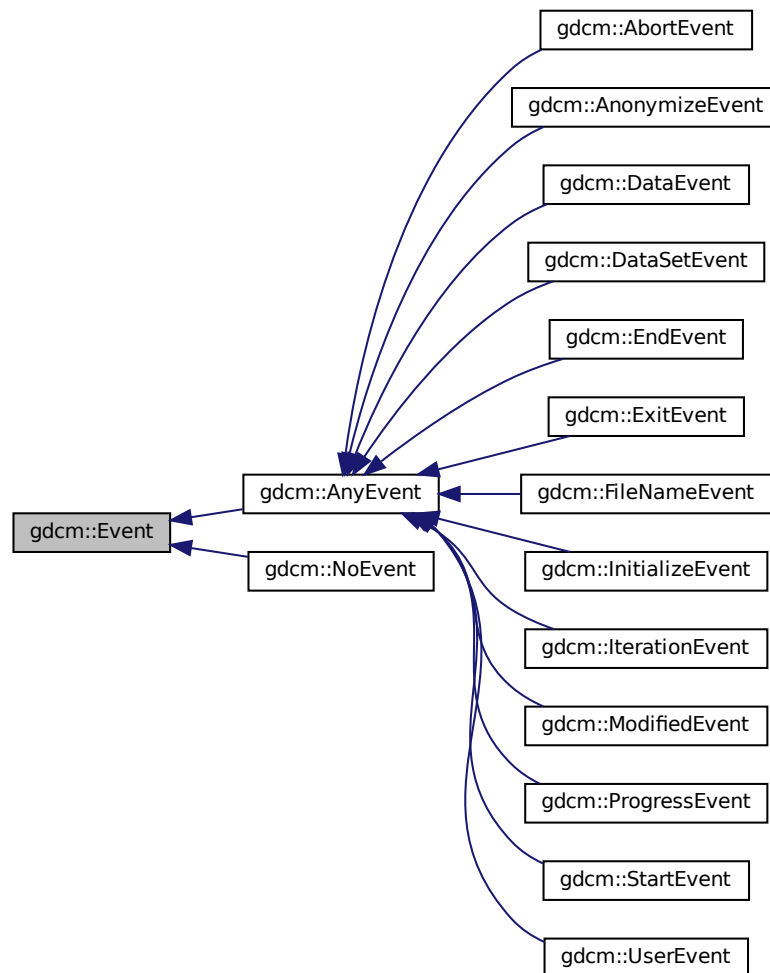
- [gdcmEnumeratedValues.h](#)

## 10.115 gdcm::Event Class Reference

superclass for callback/observer methods

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

Inheritance diagram for gdcmm::Event:



## Public Member Functions

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

### 10.115.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples:

[SimpleScanner.cxx](#).

### 10.115.2 Constructor & Destructor Documentation

#### 10.115.2.1 Event() [1/2]

```
gdcm::Event::Event ( )
```

#### 10.115.2.2 Event() [2/2]

```
gdcm::Event::Event (
    const Event & )
```

#### 10.115.2.3 ~Event()

```
virtual gdcm::Event::~~Event ( ) [virtual]
```

### 10.115.3 Member Function Documentation

#### 10.115.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
    const Event * ) const [pure virtual]
```

Check if given event matches or derives from this event.

### 10.115.3.2 GetEventName()

```
virtual const char* gdcm::Event::GetEventName (
    void ) const [pure virtual]
```

Return the StringName associated with the event.

Implemented in [gdcm::FileNameEvent](#), [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

### 10.115.3.3 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::FileNameEvent](#), [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

### 10.115.3.4 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:

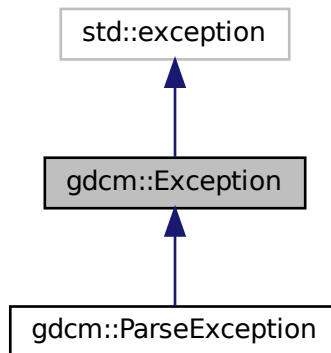
- [gdcmEvent.h](#)

## 10.116 gdcm::Exception Class Reference

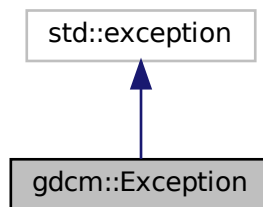
[Exception](#).

```
#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="")
- virtual [~Exception](#) () throw ()
- const char \* [GetDescription](#) () const  
*Return the Description.*
- const char \* [what](#) () const throw ()  
*what implementation*

### 10.116.1 Detailed Description

[Exception](#).

Standard exception handling object.

#### Note

Its copy-constructor and assignment operator are generated by the compiler.

### 10.116.2 Constructor & Destructor Documentation

#### 10.116.2.1 Exception()

```
gdcm::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.

#### 10.116.2.2 ~Exception()

```
virtual gdcm::Exception::~~Exception ( ) throw ( ) [inline], [virtual]
```

### 10.116.3 Member Function Documentation

#### 10.116.3.1 GetDescription()

```
const char* gdcm::Exception::GetDescription ( ) const [inline]
```

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.116.3.2 what()

```
const char* gdcm::Exception::what ( ) const throw ( )    [inline]
```

what implementation

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

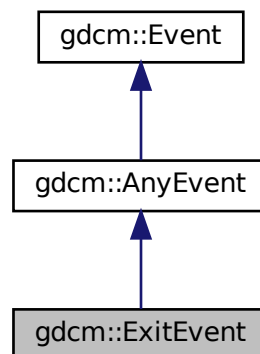
The documentation for this class was generated from the following file:

- [gdcmException.h](#)

## 10.117 gdcm::ExitEvent Class Reference

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

Inheritance diagram for `gdcm::ExitEvent`:





Collaboration diagram for gdcm::ExitEvent:



#### Additional Inherited Members

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

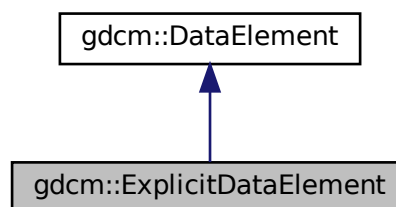
- [gdcmEvent.h](#)

## 10.118 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

## Additional Inherited Members

### 10.118.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

bla

#### Examples:

[DumpSiemensBase64.cxx](#), and [ReadAndDumpDICOMDIR2.cxx](#).

## 10.118.2 Member Function Documentation

### 10.118.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength ( ) const
```

### 10.118.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::Read (   
    std::istream & is )
```

### 10.118.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.118.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadValue (   
    std::istream & is,  
    bool readvalues = true )
```

### 10.118.2.5 ReadWithLength()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitDataElement::ReadWithLength (   
    std::istream & is,  
    VL & length )
```

### 10.118.2.6 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](#)

## 10.119 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)

## Additional Inherited Members

### 10.119.1 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.

## 10.119.2 Member Function Documentation

### 10.119.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const
```

### 10.119.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::Read (   
    std::istream & is )
```

### 10.119.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.119.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (   
    std::istream & is,   
    bool readvalues = true )
```

### 10.119.2.5 ReadWithLength()

```
template<typename TSwap >  
std::istream& gdcm::ExplicitImplicitDataElement::ReadWithLength (   
    std::istream & is,   
    VL & length ) [inline]
```

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

- [gdcmExplicitImplicitDataElement.h](#)

## 10.120 gdcm::Fiducials Class Reference

[Fiducials.](#)

```
#include <gdcmFiducials.h>
```

### Public Member Functions

- [Fiducials](#) ()

#### 10.120.1 Detailed Description

[Fiducials.](#)

#### 10.120.2 Constructor & Destructor Documentation

##### 10.120.2.1 Fiducials()

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

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

- [gdcmFiducials.h](#)

## 10.121 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](#) ()
- `const DataSet & GetDataSet () const`  
*Get Data Set.*
- `DataSet & GetDataSet ()`  
*Get Data Set.*
- `const FileMetaInformation & GetHeader () const`  
*Get [File](#) Meta Information.*
- `FileMetaInformation & GetHeader ()`  
*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.*

## Friends

- `std::ostream & operator<< (std::ostream &os, const File &val)`

## Additional Inherited Members

### 10.121.1 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](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDS\\_Explicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

### 10.121.2 Constructor & Destructor Documentation

#### 10.121.2.1 File()

```
gdcm::File::File ( )
```

### 10.121.2.2 ~File()

gdcm::File::~~File ( )

## 10.121.3 Member Function Documentation

### 10.121.3.1 GetDataSet() [1/2]

```
const DataSet& gdcm::File::GetDataSet ( ) const [inline]
```

Get Data Set.

Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), [StreamImageReaderTest.cxx](#), and [Template\\_EmptyImage.cxx](#).

### 10.121.3.2 GetDataSet() [2/2]

```
DataSet& gdcm::File::GetDataSet ( ) [inline]
```

Get Data Set.

### 10.121.3.3 GetHeader() [1/2]

```
const FileMetaInformation& gdcm::File::GetHeader ( ) const [inline]
```

Get File Meta Information.

Examples:

[CreateJPIPDataSet.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](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::operator<<()`.

#### 10.121.3.4 GetHeader() [2/2]

```
FileMetaInformation& gdcm::File::GetHeader ( ) [inline]
```

Get [File](#) Meta Information.

#### 10.121.3.5 Read()

```
std::istream& gdcm::File::Read (
    std::istream & is )
```

Read.

#### 10.121.3.6 SetDataSet()

```
void gdcm::File::SetDataSet (
    const DataSet & ds ) [inline]
```

Set Data Set.

#### 10.121.3.7 SetHeader()

```
void gdcm::File::SetHeader (
    const FileMetaInformation & fmi ) [inline]
```

Set [File](#) Meta Information.

#### 10.121.3.8 Write()

```
std::ostream const& gdcm::File::Write (
    std::ostream & os ) const
```

Write.

### 10.121.4 Friends And Related Function Documentation

#### 10.121.4.1 operator<<

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

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

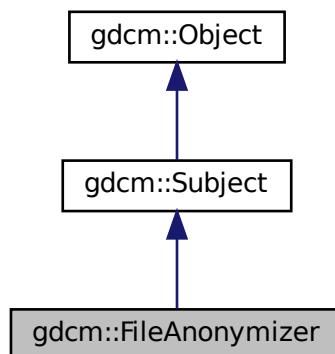
- [gdcmlFile.h](#)

### 10.122 gdcmlFileAnonymizer Class Reference

[FileAnonymizer](#).

```
#include <gdcmlFileAnonymizer.h>
```

Inheritance diagram for gdcmlFileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



### Public Member Functions

- [FileAnonymizer](#) ()
- [~FileAnonymizer](#) ()
- 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\_str)
- void [Replace](#) ([Tag](#) const &t, const char \*value\_data, [VL](#) const &vl)
- void [SetInputFileName](#) (const char \*filename\_native)  
*Set input filename.*
- void [SetOutputFileName](#) (const char \*filename\_native)  
*Set output filename.*
- bool [Write](#) ()  
*Write the output file.*

### Additional Inherited Members

#### 10.122.1 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 be 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 change 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:**

[MakeTemplate.cxx](#).

**10.122.2 Constructor & Destructor Documentation****10.122.2.1 FileAnonymizer()**

```
gdcm::FileAnonymizer::FileAnonymizer ( )
```

**10.122.2.2 ~FileAnonymizer()**

```
gdcm::FileAnonymizer::~~FileAnonymizer ( )
```

**10.122.3 Member Function Documentation****10.122.3.1 Empty()**

```
void gdcm::FileAnonymizer::Empty (
    Tag const & t )
```

Make [Tag](#) t empty **Warning:** does not handle SQ element

**Examples:**

[MakeTemplate.cxx](#).

### 10.122.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (
    Tag const & t )
```

remove a tag (even a SQ can be removed)

### 10.122.3.3 Replace() [1/2]

```
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](#) !

### 10.122.3.4 Replace() [2/2]

```
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

### 10.122.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (
    const char * filename_native )
```

Set input filename.

Examples:

[FileAnonymize.cs](#), and [MakeTemplate.cxx](#).

### 10.122.3.6 SetOutputFileName()

```
void gdcM::FileAnonymizer::SetOutputFileName (
    const char * filename_native )
```

Set output filename.

Examples:

[MakeTemplate.cxx](#).

### 10.122.3.7 Write()

```
bool gdcM::FileAnonymizer::Write ( )
```

Write the output file.

Examples:

[MakeTemplate.cxx](#).

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

- [gdcMFileAnonymizer.h](#)

## 10.123 gdcM::FileChangeTransferSyntax Class Reference

[FileChangeTransferSyntax](#).

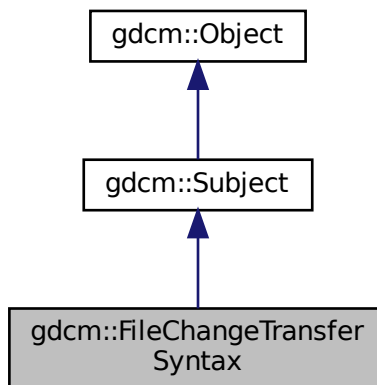
```
#include <gdcMFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcM::FileChangeTransferSyntax:





Collaboration diagram for gdcm::FileChangeTransferSyntax:



### Public Member Functions

- [FileChangeTransferSyntax](#) ()
- [~FileChangeTransferSyntax](#) ()
- [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.*

### Static Public Member Functions

- static [SmartPointer](#)< [FileChangeTransferSyntax](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

### Additional Inherited Members

#### 10.123.1 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

## 10.123.2 Constructor & Destructor Documentation

### 10.123.2.1 FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )
```

### 10.123.2.2 ~FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::~~FileChangeTransferSyntax ( )
```

## 10.123.3 Member Function Documentation

### 10.123.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ( )
```

Change the transfer syntax.

### 10.123.3.2 GetCodec()

```
ImageCodec\* gdcm::FileChangeTransferSyntax::GetCodec ( )
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

### 10.123.3.3 New()

```
static SmartPointer<FileChangeTransferSyntax> gdcm::FileChangeTransferSyntax::New ( ) [inline],  
[static]
```

for wrapped language: instantiate a reference counted object

### 10.123.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (  
    const char * filename_native )
```

Set input filename (raw DICOM)

### 10.123.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (  
    const char * filename_native )
```

Set output filename (target compressed DICOM)

### 10.123.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (  
    TransferSyntax const & ts )
```

Specify the Target Transfer Syntax.

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

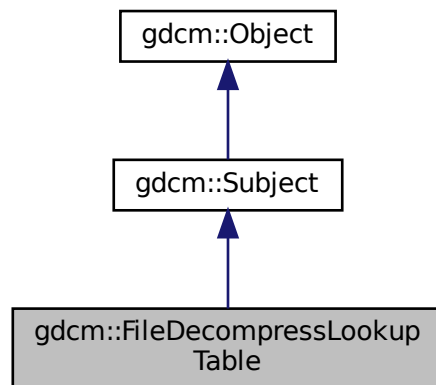
- [gdcmFileChangeTransferSyntax.h](#)

## 10.124 gdcm::FileDecompressLookupTable Class Reference

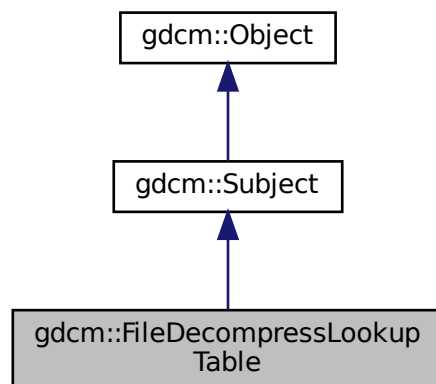
[FileDecompressLookupTable](#) class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for gdcm::FileDecompressLookupTable:



## Public Member Functions

- [FileDecompressLookupTable](#) ()
- [~FileDecompressLookupTable](#) ()
- bool [Change](#) ()  
*Decompress.*
- [File](#) & [GetFile](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const
- [Pixmap](#) & [GetPixmap](#) ()
- void [SetFile](#) (const [File](#) &f)  
*Set/Get [File](#).*
- void [SetPixmap](#) ([Pixmap](#) const &img)

## Additional Inherited Members

### 10.124.1 Detailed Description

[FileDecompressLookupTable](#) class.

It decompress the segmented LUT into linearized one (only PALETTE\_COLOR images) Output will be a [Photometric↔ Interpretation](#)=RGB image

### 10.124.2 Constructor & Destructor Documentation

#### 10.124.2.1 FileDecompressLookupTable()

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

#### 10.124.2.2 ~FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::~~FileDecompressLookupTable ( ) [inline]
```

### 10.124.3 Member Function Documentation

#### 10.124.3.1 Change()

```
bool gdcm::FileDecompressLookupTable::Change ( )
```

Decompress.

#### 10.124.3.2 GetFile()

```
File& gdcm::FileDecompressLookupTable::GetFile ( ) [inline]
```

#### 10.124.3.3 GetPixmap() [1/2]

```
const Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) const [inline]
```

#### 10.124.3.4 GetPixmap() [2/2]

```
Pixmap& gdcm::FileDecompressLookupTable::GetPixmap ( ) [inline]
```

#### 10.124.3.5 SetFile()

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

Set/Get [File](#).

#### 10.124.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
    Pixmap const & img ) [inline]
```

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

- [gdcmFileDecompressLookupTable.h](#)

## 10.125 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](#) ()

#### 10.125.1 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](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 derivation 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](#), and [GenFakelImage.cxx](#).

## 10.125.2 Constructor & Destructor Documentation

### 10.125.2.1 FileDerivation()

```
gdcm::FileDerivation::FileDerivation ( )
```

### 10.125.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~~FileDerivation ( )
```

## 10.125.3 Member Function Documentation

### 10.125.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]
```

### 10.125.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
    DataSet & ds ) [protected]
```

### 10.125.3.3 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](#), and [GenFakelImage.cxx](#).



#### 10.125.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]
```

#### 10.125.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ( )
```

Change.

Examples:

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

#### 10.125.3.6 GetFile() [1/2]

```
File& gdcm::FileDerivation::GetFile ( ) [inline]
```

Examples:

[GenFakelImage.cxx](#).

#### 10.125.3.7 GetFile() [2/2]

```
const File& gdcm::FileDerivation::GetFile ( ) const [inline]
```

#### 10.125.3.8 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.

### 10.125.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcM::FileDerivation::SetDerivationCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

#### Examples:

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

### 10.125.3.10 SetDerivationDescription()

```
void gdcM::FileDerivation::SetDerivationDescription (
    const char * dd )
```

Specify the Derivation Description. Eg "lossy conversion".

### 10.125.3.11 SetFile()

```
void gdcM::FileDerivation::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

#### Examples:

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

### 10.125.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcM::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
    unsigned int codevalue )
```

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

#### Examples:

[DeriveSeries.cxx](#), and [GenFakelImage.cxx](#).

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

- [gdcMFileDerivation.h](#)

## 10.126 gdcm::FileExplicitFilter Class Reference

[FileExplicitFilter](#) class.

```
#include <gdcmFileExplicitFilter.h>
```

### Public Member Functions

- [FileExplicitFilter](#) ()
- [~FileExplicitFilter](#) ()
- 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)

#### 10.126.1 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:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

## 10.126.2 Constructor & Destructor Documentation

### 10.126.2.1 FileExplicitFilter()

```
gdcM::FileExplicitFilter::FileExplicitFilter ( ) [inline]
```

### 10.126.2.2 ~FileExplicitFilter()

```
gdcM::FileExplicitFilter::~~FileExplicitFilter ( ) [inline]
```

## 10.126.3 Member Function Documentation

### 10.126.3.1 Change()

```
bool gdcM::FileExplicitFilter::Change ( )
```

Set FMI Transfer Syntax.

Change

Examples:

[GenAIIVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

### 10.126.3.2 ChangeFMI()

```
bool gdcM::FileExplicitFilter::ChangeFMI ( ) [protected]
```

### 10.126.3.3 GetFile()

```
File& gdcM::FileExplicitFilter::GetFile ( ) [inline]
```

#### 10.126.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
    DataSet & ds,
    Dicts const & dicts ) [protected]
```

#### 10.126.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
    bool b ) [inline]
```

Decide whether or not to [VR](#)ify private tags.

#### 10.126.3.6 SetFile()

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

Set/Get [File](#).

#### Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

#### 10.126.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
    bool b )
```

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

#### 10.126.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
    bool b )
```

### 10.126.3.9 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:

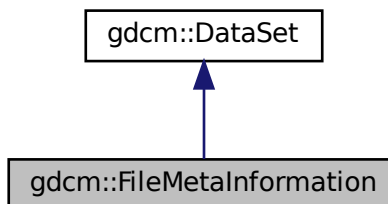
- [gdcmFileExplicitFilter.h](#)

## 10.127 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)
- [~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
- const [Preamble](#) & [GetPreamble](#) () const  
*Get [Preamble](#).*
- [Preamble](#) & [GetPreamble](#) ()
- void [Insert](#) (const [DataElement](#) &de)
- bool [IsValid](#) () const
- 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.*

## 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)

## 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

### 10.127.1 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:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

### 10.127.2 Constructor & Destructor Documentation



### 10.127.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ( )
```

### 10.127.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~~FileMetaInformation ( )
```

### 10.127.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
    FileMetaInformation const & fmi ) [inline]
```

References DataSetMS, DataSetTS, and MetaInformationTS.

## 10.127.3 Member Function Documentation

### 10.127.3.1 AppendImplementationClassUID()

```
static void gdcm::FileMetaInformation::AppendImplementationClassUID (
    const char * imp ) [static]
```

### 10.127.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]
```

### 10.127.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]
```

#### 10.127.3.4 Default()

```
void gdcm::FileMetaInformation::Default ( ) [protected]
```

#### 10.127.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
    DataSet const & ds )
```

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

#### 10.127.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]
```

Examples:

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

#### 10.127.3.7 GetFileMetaInformationVersion()

```
static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]
```

#### 10.127.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]
```

References [gdcm::VL::GetLength\(\)](#).

#### 10.127.3.9 GetGDCMImplementationClassUID()

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]
```

**10.127.3.10 GetGDCMImplementationVersionName()**

```
static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static],  
[protected]
```

**10.127.3.11 GetGDCMSourceApplicationEntityTitle()**

```
static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static],  
[protected]
```

**10.127.3.12 GetImplementationClassUID()**

```
static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]
```

**10.127.3.13 GetImplementationVersionName()**

```
static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]
```

**10.127.3.14 GetMediaStorage()**

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const
```

**10.127.3.15 GetMediaStorageAsString()**

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const
```

**10.127.3.16 GetMetaInformationTS()**

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]
```

**10.127.3.17 GetPreamble()** [1/2]

```
const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]
```

Get [Preamble](#).

Referenced by `gdcm::operator<<()`.

**10.127.3.18 GetPreamble()** [2/2]

```
Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]
```

**10.127.3.19 GetSourceApplicationEntityTitle()**

```
static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]
```

**10.127.3.20 Insert()**

```
void gdcm::FileMetaInformation::Insert (
    const DataElement & de ) [inline]
```

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

**10.127.3.21 IsValid()**

```
bool gdcm::FileMetaInformation::IsValid ( ) const [inline]
```

**10.127.3.22 Read()**

```
std::istream& gdcm::FileMetaInformation::Read (
    std::istream & is )
```

Read.

### 10.127.3.23 ReadCompat()

```
std::istream& gdcm::FileMetaInformation::ReadCompat (
    std::istream & is )
```

### 10.127.3.24 ReadCompatInternal()

```
template<typename TSwap >
std::istream& gdcm::FileMetaInformation::ReadCompatInternal (
    std::istream & is ) [protected]
```

### 10.127.3.25 Replace()

```
void gdcm::FileMetaInformation::Replace (
    const DataElement & de ) [inline]
```

#### Examples:

[LargeVRDSExplicit.cxx](#).

References [gdcm::DataElement::GetTag\(\)](#).

### 10.127.3.26 SetDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::SetDataSetTransferSyntax (
    const TransferSyntax & ts )
```

#### Examples:

[CreateJPIPDataSet.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](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

#### 10.127.3.27 SetImplementationClassUID()

```
static void gdcm::FileMetaInformation::SetImplementationClassUID (
    const char * imp ) [static]
```

Override the GDCM default values:

#### 10.127.3.28 SetImplementationVersionName()

```
static void gdcm::FileMetaInformation::SetImplementationVersionName (
    const char * version ) [static]
```

#### 10.127.3.29 SetPreamble()

```
void gdcm::FileMetaInformation::SetPreamble (
    const Preamble & p ) [inline]
```

#### 10.127.3.30 SetSourceApplicationEntityTitle()

```
static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (
    const char * title ) [static]
```

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [FixJAIBugJPEGLS.cxx](#), [GenerateDICOMDIR.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

#### 10.127.3.31 Write()

```
std::ostream& gdcm::FileMetaInformation::Write (
    std::ostream & os ) const
```

Write.

### 10.127.4 Friends And Related Function Documentation

#### 10.127.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const FileMetaInformation & _val ) [friend]
```

### 10.127.5 Member Data Documentation

#### 10.127.5.1 DataSetMS

[MediaStorage::MSType](#) gdcm::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

#### 10.127.5.2 DataSetTS

[TransferSyntax](#) gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

#### 10.127.5.3 MetaInformationTS

[TransferSyntax::NegociatedType](#) gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- [gdcmFileMetaInformation.h](#)

## 10.128 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)

### 10.128.1 Detailed Description

Class to manipulate file name's.

#### Note

OS independant representation of a filename (to query path, name and extension from a filename)

### 10.128.2 Constructor & Destructor Documentation

#### 10.128.2.1 [Filename](#)()

```
gdcm::Filename::Filename (
    const char * filename = "" ) [inline]
```



### 10.128.3 Member Function Documentation

#### 10.128.3.1 EndWith()

```
bool gdcm::Filename::EndWith (
    const char ending[] ) const
```

Does the filename ends with a particular string ?

#### 10.128.3.2 GetExtension()

```
const char* gdcm::Filename::GetExtension ( )
```

return only the extension part of a filename

#### 10.128.3.3 GetFileName()

```
const char* gdcm::Filename::GetFileName ( ) const [inline]
```

Return the full filename.

#### 10.128.3.4 GetName()

```
const char* gdcm::Filename::GetName ( )
```

return only the name part of a filename

#### 10.128.3.5 GetPath()

```
const char* gdcm::Filename::GetPath ( )
```

Return only the path component of a filename.

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#).

### 10.128.3.6 IsEmpty()

```
bool gdcM::Filename::IsEmpty ( ) const [inline]
```

return whether the filename is empty

### 10.128.3.7 IsIdentical()

```
bool gdcM::Filename::IsIdentical (
    Filename const & fn ) const
```

### 10.128.3.8 Join()

```
static const char* gdcM::Filename::Join (
    const char * path,
    const char * filename ) [static]
```

Join two paths NOT THREAD SAFE

Examples:

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

### 10.128.3.9 operator const char \*()

```
gdcM::Filename::operator const char * ( ) const [inline]
```

Simple operator to allow [Filename](#) myfilename( "..."); const char \* s = myfilename;

### 10.128.3.10 ToUnixSlashes()

```
const char* gdcM::Filename::ToUnixSlashes ( )
```

Convert backslash (windows style) to UNIX style slash.

## 10.128.3.11 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](#)

## 10.129 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`)
- virtual `~FileNameEvent` ()
- virtual `bool CheckEvent` (`const ::gdcm::Event *e`) `const`
- virtual `const char * GetEventName` () `const`
- `const char * GetFileName` () `const`
- virtual `::gdcm::Event * MakeObject` () `const`
- void `SetFileName` (`const char *f`)

### 10.129.1 Detailed Description

`FileNameEvent`.

Special type of event triggered during processing of `FileSet`

See also

[`AnyEvent`](#)

Examples:

[`SimpleScanner.cxx`](#).

## 10.129.2 Member Typedef Documentation

### 10.129.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

### 10.129.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

## 10.129.3 Constructor & Destructor Documentation

### 10.129.3.1 [FileNameEvent\(\)](#) [1/2]

```
gdcm::FileNameEvent::FileNameEvent (  
    const char * s = "" ) [inline]
```

### 10.129.3.2 [~FileNameEvent\(\)](#)

```
virtual gdcm::FileNameEvent::~~FileNameEvent ( ) [inline], [virtual]
```

### 10.129.3.3 [FileNameEvent\(\)](#) [2/2]

```
gdcm::FileNameEvent::FileNameEvent (  
    const Self & s ) [inline]
```

## 10.129.4 Member Function Documentation

#### 10.129.4.1 CheckEvent()

```
virtual bool gdcm::FileNameEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [virtual]
```

#### 10.129.4.2 GetEventName()

```
virtual const char* gdcm::FileNameEvent::GetEventName ( ) const [inline], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

#### 10.129.4.3 GetFileName()

```
const char* gdcm::FileNameEvent::GetFileName ( ) const [inline]
```

Examples:

[SimpleScanner.cxx](#).

#### 10.129.4.4 MakeObject()

```
virtual ::gdcm::Event* gdcm::FileNameEvent::MakeObject ( ) const [inline], [virtual]
```

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

Implements [gdcm::Event](#).

#### 10.129.4.5 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
    const char * f ) [inline]
```

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

- [gdcmFileNameEvent.h](#)

## 10.130 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](#) ()
- 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.*

#### 10.130.1 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](#).

## 10.130.2 Member Typedef Documentation

### 10.130.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType
```

### 10.130.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

### 10.130.2.3 SizeType

```
typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType
```

## 10.130.3 Constructor & Destructor Documentation

### 10.130.3.1 FilenameGenerator()

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

### 10.130.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~~FilenameGenerator ( ) [inline]
```

## 10.130.4 Member Function Documentation



#### 10.130.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ( )
```

Generate (return success)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.130.4.2 GetFilename()

```
const char* gdcm::FilenameGenerator::GetFilename (
    SizeType n ) const
```

Get a particular filename (call after Generate)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.130.4.3 GetFileNames()

```
FileNamesType const& gdcm::FilenameGenerator::GetFileNames ( ) const [inline]
```

Return all filenames.

#### 10.130.4.4 GetNumberOfFileNames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFileNames ( ) const
```

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.130.4.5 GetPattern()

```
const char* gdcm::FilenameGenerator::GetPattern ( ) const [inline]
```

#### 10.130.4.6 GetPrefix()

```
const char* gdcm::FilenameGenerator::GetPrefix ( ) const [inline]
```

#### 10.130.4.7 SetNumberOfFileNames()

```
void gdcm::FilenameGenerator::SetNumberOfFileNames (
    SizeType nfiles )
```

Set/Get the number of filenames to generate.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.130.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
    const char * pattern ) [inline]
```

Set/Get pattern.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.130.4.9 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](#)

## 10.131 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

### Public Types

- typedef std::vector< [FileType](#) > [FilesType](#)
- typedef std::string [FileType](#)

### Public Member Functions

- [FileSet](#) ()
- void [AddFile](#) ([File](#) const &)
- bool [AddFile](#) (const char \*filename)
- [FilesType](#) const & [GetFiles](#) () const
- void [SetFiles](#) ([FilesType](#) const &files)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [FileSet](#) &d)

#### 10.131.1 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.

#### 10.131.2 Member Typedef Documentation

##### 10.131.2.1 FilesType

```
typedef std::vector<FileType> gdcm::FileSet::FilesType
```

##### 10.131.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

### 10.131.3 Constructor & Destructor Documentation

#### 10.131.3.1 FileSet()

```
gdcmm::FileSet::FileSet ( ) [inline]
```

### 10.131.4 Member Function Documentation

#### 10.131.4.1 AddFile() [1/2]

```
void gdcmm::FileSet::AddFile (
    File const & ) [inline]
```

**Deprecated** . Does nothing

#### 10.131.4.2 AddFile() [2/2]

```
bool gdcmm::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.

#### 10.131.4.3 GetFiles()

```
FileType const& gdcmm::FileSet::GetFiles ( ) const [inline]
```

#### 10.131.4.4 SetFiles()

```
void gdcmm::FileSet::SetFiles (
    FileType const & files )
```

### 10.131.5 Friends And Related Function Documentation

#### 10.131.5.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & _os,  
    const FileSet & d ) [friend]
```

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

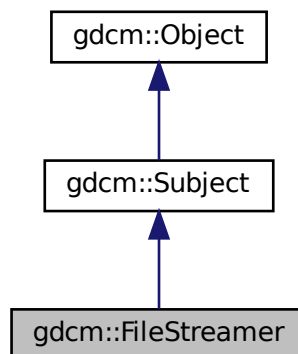
- [gdcmFileSet.h](#)

## 10.132 gdcm::FileStreamer Class Reference

[FileStreamer.](#)

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for `gdcm::FileStreamer`:



## Public Member Functions

- [FileStreamer](#) ()
- [~FileStreamer](#) ()
- 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.*

## Static Public Member Functions

- static [SmartPointer](#)< [FileStreamer](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

### 10.132.1 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.

### 10.132.2 Constructor & Destructor Documentation

#### 10.132.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ( )
```

#### 10.132.2.2 ~FileStreamer()

```
gdcm::FileStreamer::~~FileStreamer ( )
```

### 10.132.3 Member Function Documentation

#### 10.132.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
    const Tag & t,
    const char * array,
    size_t len )
```

Append to previously started [Tag](#) t.

### 10.132.3.2 AppendToGroupDataElement()

```
bool gdcM::FileStreamer::AppendToGroupDataElement (
    const PrivateTag & pt,
    const char * array,
    size_t len )
```

Append to previously started private creator.

### 10.132.3.3 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

### 10.132.3.4 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.

### 10.132.3.5 New()

```
static SmartPointer<FileStreamer> gdcM::FileStreamer::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

### 10.132.3.6 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



### 10.132.3.7 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

### 10.132.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
    const char * filename_native )
```

Set output filename (target file)

### 10.132.3.9 SetTemplateFileName()

```
void gdcm::FileStreamer::SetTemplateFileName (
    const char * filename_native )
```

Set input DICOM template filename.

#### Examples:

[FileStreaming.cs](#).

### 10.132.3.10 StartDataElement()

```
bool gdcm::FileStreamer::StartDataElement (
    const Tag & t )
```

Start Single Data Element Operation This will delete any existing Tag t. Need to call it only once.

### 10.132.3.11 StartGroupDataElement()

```
bool gdcm::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

<i>maxsize</i>	. 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). <a href="#">Value</a> should be in [0x0, 0xff] This will find the first available private creator.
----------------	--

## 10.132.3.12 StopDataElement()

```
bool gdcm::FileStreamer::StopDataElement (
    const Tag & t )
```

Stop appending to tag t. This will compute the proper attribute length.

## 10.132.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (
    const PrivateTag & pt )
```

Stop appending to private creator.

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

- [gdcmFileStreamer.h](#)

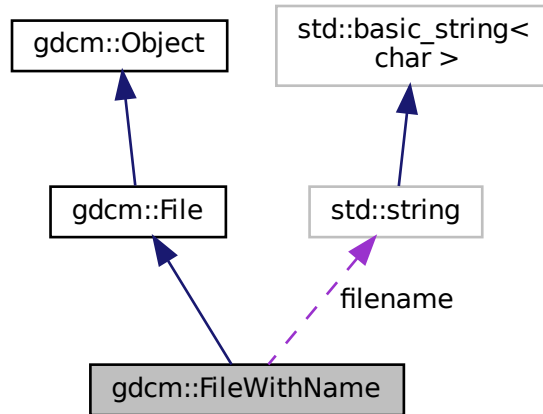
## 10.133 gdcm::FileWithName Class Reference

[FileWithName.](#)

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:





- `FileWithName (File &f)`

- `std::string filename`

## FileWithName.

Backward only class do not use in newer code

### 10.133.2.1 FileWithName()

```
gdcmm::FileWithName::FileWithName (
    File & f ) [inline]
```

## 10.133.3 Member Data Documentation

### 10.133.3.1 filename

```
std::string gdcmm::FileWithName::filename
```

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

- [gdcmmSerieHelper.h](#)

## 10.134 gdcmm::FindPatientRootQuery Class Reference

PatientRootQuery.

```
#include <gdcmmFindPatientRootQuery.h>
```

Inheritance diagram for gdcmm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



## Public Member Functions

- [FindPatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.134.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

### 10.134.2 Constructor & Destructor Documentation

#### 10.134.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ( )
```

### 10.134.3 Member Function Documentation

#### 10.134.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.134.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::FindPatientRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [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](#).

#### 10.134.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [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 dcm4k

Implements [gdcm::BaseRootQuery](#).

#### 10.134.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (
    bool inStrict = true ) const [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](#).

#### 10.134.4 Friends And Related Function Documentation

##### 10.134.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

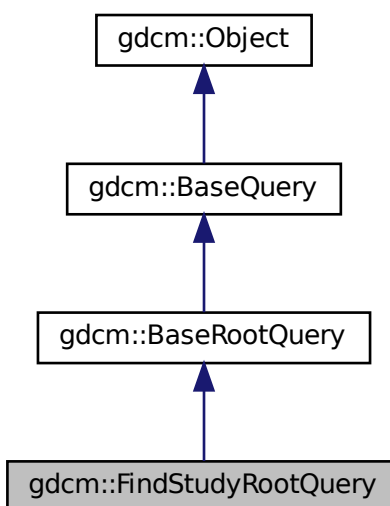
- [gdcmFindPatientRootQuery.h](#)

### 10.135 gdcm::FindStudyRootQuery Class Reference

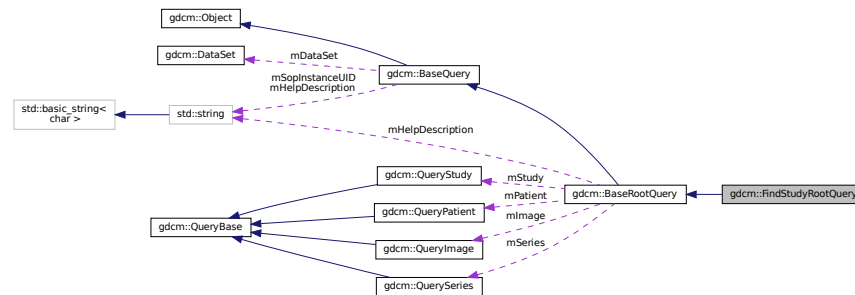
[FindStudyRootQuery](#).

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for `gdcm::FindStudyRootQuery`:



## Public Member Functions

- [FindStudyRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.135.1 Detailed Description

[FindStudyRootQuery](#).

contains: the class which will produce a dataset for C-FIND with study root

### 10.135.2 Constructor & Destructor Documentation

#### 10.135.2.1 FindStudyRootQuery()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ( )
```



### 10.135.3 Member Function Documentation

#### 10.135.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.135.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::FindStudyRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [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](#).

#### 10.135.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [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 dcmrk

Implements [gdcm::BaseRootQuery](#).

#### 10.135.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (
    bool inStrict = true ) const [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](#).

## 10.135.4 Friends And Related Function Documentation

### 10.135.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

- [gdcmmFindStudyRootQuery.h](#)

## 10.136 gdcmm::Fragment Class Reference

Class to represent a [Fragment](#).

```
#include <gdcmmFragment.h>
```

Inheritance diagram for gdcmm::Fragment:



Collaboration diagram for gdcm::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

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Fragment](#) &val)

## Additional Inherited Members

### 10.136.1 Detailed Description

Class to represent a [Fragment](#).

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

## 10.136.2 Constructor & Destructor Documentation

### 10.136.2.1 Fragment()

```
gdcM::Fragment::Fragment ( ) [inline]
```

References `gdcM::operator<<()`.

## 10.136.3 Member Function Documentation

### 10.136.3.1 ComputeLength()

```
VL gdcM::Fragment::ComputeLength ( ) const
```

### 10.136.3.2 GetLength()

```
VL gdcM::Fragment::GetLength ( ) const
```

### 10.136.3.3 Read()

```
template<typename TSwap >  
std::istream& gdcM::Fragment::Read (  
    std::istream & is ) [inline]
```

Referenced by `gdcM::SequenceOfFragments::ReadValue()`.

### 10.136.3.4 ReadBacktrack()

```
template<typename TSwap >  
std::istream& gdcM::Fragment::ReadBacktrack (  
    std::istream & is ) [inline]
```

References `gdcMErrorMacro`, `gdcMWarningMacro`, and `gdcM::ParseException::SetLastElement()`.

Referenced by `gdcM::SequenceOfFragments::ReadValue()`.

### 10.136.3.5 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadPreValue (
    std::istream & is ) [inline]
```

### 10.136.3.6 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::Fragment::ReadValue (
    std::istream & is ) [inline]
```

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

### 10.136.3.7 Write()

```
template<typename TSwap >
std::ostream& gdcm::Fragment::Write (
    std::ostream & os ) const [inline]
```

References `gdcm::ByteValue::ComputeLength()`, `gdcm::ByteValue::GetLength()`, `gdcm::VL::Write()`, and `gdcm::ByteValue::Write()`.

## 10.136.4 Friends And Related Function Documentation

### 10.136.4.1 operator<<

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

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

- [gdcmFragment.h](#)

## 10.137 gdcm::Global Class Reference

[Global](#).

```
#include <gdcmGlobal.h>
```

## Public Member Functions

- [Global](#) ()
- [~Global](#) ()
- bool [Append](#) (const char \*path)
- [Defs](#) const & [GetDefs](#) () const
- [Dicts](#) const & [GetDicts](#) () const
- [Dicts](#) & [GetDicts](#) ()
- bool [LoadResourcesFiles](#) ()
- 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)

## 10.137.1 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:

[GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

## 10.137.2 Constructor & Destructor Documentation

### 10.137.2.1 Global()

```
gdcm::Global::Global ( )
```

### 10.137.2.2 ~Global()

```
gdcm::Global::~~Global ( )
```

## 10.137.3 Member Function Documentation

### 10.137.3.1 Append()

```
bool gdcm::Global::Append (
    const char * path )
```

Append path at the end of the path list

#### Warning

not thread safe !

### 10.137.3.2 GetDefs()

```
Defs const& gdcm::Global::GetDefs ( ) const
```

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

#### Examples:

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

### 10.137.3.3 GetDicts() [1/2]

```
Dicts const& gdcm::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](#).

### 10.137.3.4 GetDicts() [2/2]

```
Dicts& gdcm::Global::GetDicts ( )
```

### 10.137.3.5 GetInstance()

```
static Global& gdcm::Global::GetInstance ( ) [static]
```

return the singleton instance

Examples:

[BasicAnonymizer.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

### 10.137.3.6 LoadResourcesFiles()

```
bool gdcm::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:

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



### 10.137.3.7 Locate()

```
const char* gdcm::Global::Locate (
    const char * resfile ) const [protected]
```

Locate a resource file.

### 10.137.3.8 Prepend()

```
bool gdcm::Global::Prepend (
    const char * path )
```

Prepend path at the beginning of the path list

#### Warning

not thread safe !

## 10.137.4 Friends And Related Function Documentation

### 10.137.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Global & g ) [friend]
```

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

- [gdcmGlobal.h](#)

## 10.138 gdcm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmGroupDict.h>
```

### Public Types

- typedef std::vector< std::string > [GroupStringVector](#)

## Public Member Functions

- [GroupDict](#) ()
- [~GroupDict](#) ()
- 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)

### 10.138.1 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 ?

### 10.138.2 Member Typedef Documentation

#### 10.138.2.1 GroupStringVector

```
typedef std::vector<std::string> gdc::GroupDict::GroupStringVector
```

### 10.138.3 Constructor & Destructor Documentation

#### 10.138.3.1 GroupDict()

```
gdc::GroupDict::GroupDict ( ) [inline]
```

### 10.138.3.2 ~GroupDict()

```
gdcm::GroupDict::~~GroupDict ( ) [inline]
```

References `gdcm::operator<<()`.

## 10.138.4 Member Function Documentation

### 10.138.4.1 Add()

```
void gdcm::GroupDict::Add (
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

### 10.138.4.2 GetAbbreviation()

```
std::string const& gdcm::GroupDict::GetAbbreviation (
    uint16_t num ) const
```

Referenced by `gdcm::operator<<()`.

### 10.138.4.3 GetName()

```
std::string const& gdcm::GroupDict::GetName (
    uint16_t num ) const
```

Referenced by `gdcm::operator<<()`.

### 10.138.4.4 Insert()

```
void gdcm::GroupDict::Insert (
    uint16_t num,
    std::string const & abbreviation,
    std::string const & name ) [protected]
```

#### 10.138.4.5 Size()

```
size_t gdcmm::GroupDict::Size ( ) const [inline]
```

Referenced by `gdcmm::operator<<()`.

### 10.138.5 Friends And Related Function Documentation

#### 10.138.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const GroupDict & _val ) [friend]
```

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

- [gdcmmGroupDict.h](#)

## 10.139 gdcmm::IconImageFilter Class Reference

[IconImageFilter](#).

```
#include <gdcmmIconImageFilter.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](#) ()

### 10.139.1 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](#).

### 10.139.2 Constructor & Destructor Documentation

#### 10.139.2.1 [IconImageFilter](#)()

```
gdcm::IconImageFilter::IconImageFilter ( )
```

#### 10.139.2.2 [~IconImageFilter](#)()

```
gdcm::IconImageFilter::~~IconImageFilter ( )
```

### 10.139.3 Member Function Documentation

#### 10.139.3.1 Extract()

```
bool gdcM::IconImageFilter::Extract ( )
```

Extract all Icon found in [File](#).

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.139.3.2 ExtractIconImages()

```
void gdcM::IconImageFilter::ExtractIconImages ( ) [protected]
```

#### 10.139.3.3 ExtractVeprolIconImages()

```
void gdcM::IconImageFilter::ExtractVeproIconImages ( ) [protected]
```

#### 10.139.3.4 GetFile() [1/2]

```
File& gdcM::IconImageFilter::GetFile ( ) [inline]
```

#### 10.139.3.5 GetFile() [2/2]

```
const File& gdcM::IconImageFilter::GetFile ( ) const [inline]
```

### 10.139.3.6 GetIconImage()

```
IconImage& gdcm::IconImageFilter::GetIconImage (
    unsigned int i ) const
```

Examples:

[ExtractIconFromFile.cxx](#).

### 10.139.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const
```

Retrieve extract IconImage (need to call Extract first)

Examples:

[ExtractIconFromFile.cxx](#).

### 10.139.3.8 SetFile()

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

Set/Get [File](#).

Examples:

[ExtractIconFromFile.cxx](#).

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

- [gdcmIconImageFilter.h](#)

## 10.140 gdcm::IconImageGenerator Class Reference

[IconImageGenerator](#).

```
#include <gdcmIconImageGenerator.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.*

### 10.140.1 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](#).

### 10.140.2 Constructor & Destructor Documentation



### 10.140.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ( )
```

### 10.140.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~~IconImageGenerator ( )
```

## 10.140.3 Member Function Documentation

### 10.140.3.1 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](#).

### 10.140.3.2 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

### 10.140.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ( )
```

Generate Icon.

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.140.3.4 GetIconImage()

```
const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const [inline]
```

Retrieve generated Icon.

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.140.3.5 GetPixmap() [1/2]

```
Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) [inline]
```

#### 10.140.3.6 GetPixmap() [2/2]

```
const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const [inline]
```

#### 10.140.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
    const unsigned int dims[2] )
```

Set Target dimension of output Icon.

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.140.3.8 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)`

#### 10.140.3.9 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.

#### 10.140.3.10 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](#)

## 10.141 gdcm::ignore\_char Struct Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- [ignore\\_char](#) (char c)

### Public Attributes

- char [m\\_char](#)

#### 10.141.1 Constructor & Destructor Documentation

#### 10.141.1.1 ignore\_char()

```
gdcmm::ignore_char::ignore_char (  
    char c )    [inline]
```

### 10.141.2 Member Data Documentation

#### 10.141.2.1 m\_char

```
char gdcmm::ignore_char::m_char
```

Referenced by `gdcmm::operator>>()`.

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

- [gdcmmElement.h](#)

## 10.142 gdcmm::Image Class Reference

[Image](#).

```
#include <gdcmmImage.h>
```

Inheritance diagram for `gdcmm::Image`:





- `Image ()`
  - `~Image ()`
  - `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`
- print*
- `void SetDirectionCosines (const float *dircos)`
  - `void SetDirectionCosines (const double *dircos)`
  - `void SetDirectionCosines (unsigned int idx, double dircos)`
  - `void SetIntercept (double intercept)`
- intercept*
- `void SetOrigin (const float *ori)`
  - `void SetOrigin (const double *ori)`
  - `void SetOrigin (unsigned int idx, double ori)`
  - `void SetSlope (double slope)`
- slope*
- `void SetSpacing (const double *spacing)`
  - `void SetSpacing (unsigned int idx, double spacing)`

## Additional Inherited Members

### 10.142.1 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 seems 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:

[CompressImage.cxx](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAI BugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

### 10.142.2 Constructor & Destructor Documentation

#### 10.142.2.1 Image()

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

### 10.142.2.2 ~Image()

```
gdcm::Image::~~Image ( ) [inline]
```

## 10.142.3 Member Function Documentation

### 10.142.3.1 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.

### 10.142.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
    unsigned int idx ) const
```

### 10.142.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept ( ) const [inline]
```

### 10.142.3.4 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](#).

**10.142.3.5 GetOrigin()** [2/2]

```
double gdcM::Image::GetOrigin (
    unsigned int idx ) const
```

**10.142.3.6 GetSlope()**

```
double gdcM::Image::GetSlope ( ) const [inline]
```

**10.142.3.7 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

**10.142.3.8 GetSpacing()** [2/2]

```
double gdcM::Image::GetSpacing (
    unsigned int idx ) const
```

**10.142.3.9 Print()**

```
void gdcM::Image::Print (
    std::ostream & os ) const [virtual]
```

print

Reimplemented from [gdcM::Bitmap](#).

Examples:

[CompressImage.cxx](#), and [PatchFile.cxx](#).



**10.142.3.10 SetDirectionCosines()** [1/3]

```
void gdcm::Image::SetDirectionCosines (
    const float * dircos )
```

**10.142.3.11 SetDirectionCosines()** [2/3]

```
void gdcm::Image::SetDirectionCosines (
    const double * dircos )
```

**10.142.3.12 SetDirectionCosines()** [3/3]

```
void gdcm::Image::SetDirectionCosines (
    unsigned int idx,
    double dircos )
```

**10.142.3.13 SetIntercept()**

```
void gdcm::Image::SetIntercept (
    double intercept ) [inline]
```

*intercept*

**Examples:**

[TemplateEmptyImage.cxx](#).

**10.142.3.14 SetOrigin()** [1/3]

```
void gdcm::Image::SetOrigin (
    const float * ori )
```

**10.142.3.15 SetOrigin()** [2/3]

```
void gdcM::Image::SetOrigin (
    const double * ori )
```

**10.142.3.16 SetOrigin()** [3/3]

```
void gdcM::Image::SetOrigin (
    unsigned int idx,
    double ori )
```

**10.142.3.17 SetSlope()**

```
void gdcM::Image::SetSlope (
    double slope ) [inline]
```

slope

Examples:

[TemplateEmptyImage.cxx](#).

**10.142.3.18 SetSpacing()** [1/2]

```
void gdcM::Image::SetSpacing (
    const double * spacing )
```

Examples:

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

**10.142.3.19 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](#)

## 10.143 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:*

### Additional Inherited Members

### 10.143.1 Detailed Description

[ImageApplyLookupTable](#) class.

It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image

### 10.143.2 Constructor & Destructor Documentation

#### 10.143.2.1 ImageApplyLookupTable()

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

#### 10.143.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ( ) [inline]
```

### 10.143.3 Member Function Documentation

#### 10.143.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ( )
```

Apply.

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

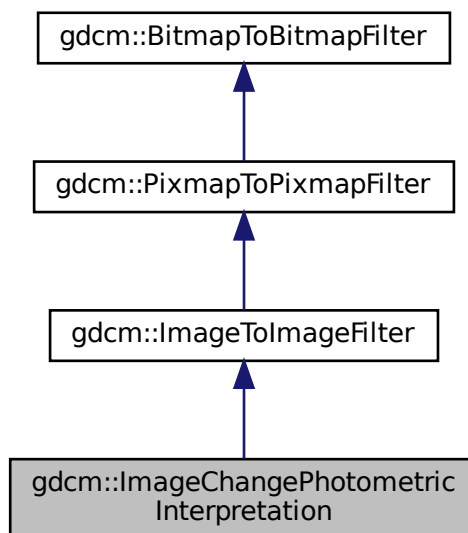
- [gdcmImageApplyLookupTable.h](#)

## 10.144 gdcm::ImageChangePhotometricInterpretation Class Reference

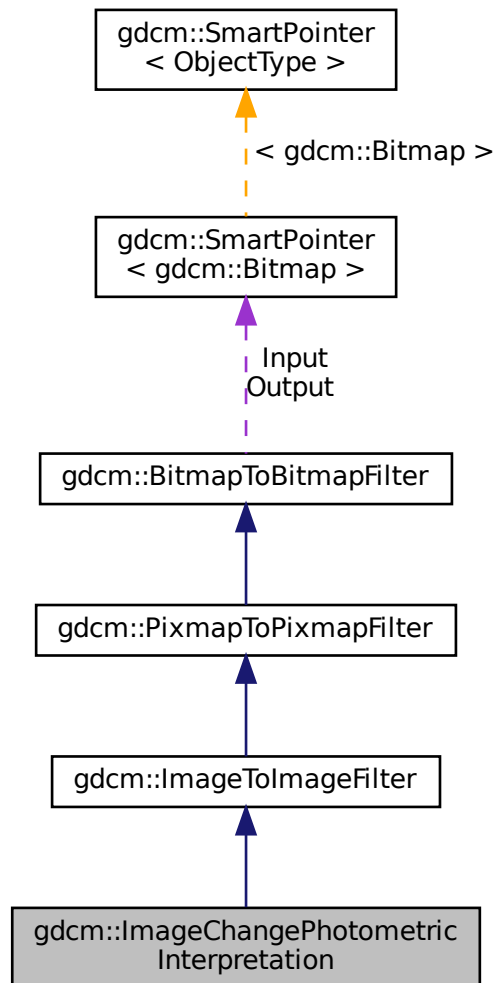
[ImageChangePhotometricInterpretation](#) class.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



## Public Member Functions

- [ImageChangePhotometricInterpretation](#) ()
- [~ImageChangePhotometricInterpretation](#) ()
- [bool Change](#) ()  
*Change.*
- [const PhotometricInterpretation & GetPhotometricInterpretation](#) () const
- [void SetPhotometricInterpretation](#) ([PhotometricInterpretation](#) const &pi)  
*Set/Get requested PhotometricInterpretation.*

## Static Public Member Functions

- `template<typename T >`  
static void [RGB2YBR](#) (T ybr[3], const T rgb[3])
- `template<typename T >`  
static void [YBR2RGB](#) (T rgb[3], const T ybr[3])

## Protected Member Functions

- bool [ChangeMonochrome](#) ()
- bool [ChangeRGB2YBR](#) ()
- bool [ChangeYBR2RGB](#) ()

## Additional Inherited Members

### 10.144.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class.

Class to change the Photometric Interpretation of an input DICOM

### 10.144.2 Constructor & Destructor Documentation

#### 10.144.2.1 [ImageChangePhotometricInterpretation\(\)](#)

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

#### 10.144.2.2 [~ImageChangePhotometricInterpretation\(\)](#)

```
gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ( ) [inline]
```

### 10.144.3 Member Function Documentation



### 10.144.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ( )
```

Change.

### 10.144.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]
```

### 10.144.3.3 ChangeRGB2YBR()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR ( ) [protected]
```

### 10.144.3.4 ChangeYBR2RGB()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB ( ) [protected]
```

### 10.144.3.5 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometric↵  
Interpretation ( ) const [inline]
```

### 10.144.3.6 RGB2YBR()

```
template<typename T >  
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (   
    T ybr[3],  
    const T rgb[3] ) [static]
```

colorspace conversion (based on CCIR Recommendation 601-2) -> T.871

References gdcm::Clamp(), and gdcm::Round().

### 10.144.3.7 SetPhotometricInterpretation()

```
void gdcM::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi ) [inline]
```

Set/Get requested [PhotometricInterpretation](#).

### 10.144.3.8 YBR2RGB()

```
template<typename T >
void gdcM::ImageChangePhotometricInterpretation::YBR2RGB (
    T rgb[3],
    const T ybr[3] ) [static]
```

References `gdcM::Clamp()`, and `gdcM::Round()`.

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

- [gdcMImageChangePhotometricInterpretation.h](#)

## 10.145 gdcM::ImageChangePlanarConfiguration Class Reference

[ImageChangePlanarConfiguration](#) class.

```
#include <gdcMImageChangePlanarConfiguration.h>
```

Inheritance diagram for `gdcM::ImageChangePlanarConfiguration`:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



## Public Member Functions

- [ImageChangePlanarConfiguration](#) ()
- [~ImageChangePlanarConfiguration](#) ()
- bool [Change](#) ()  
*Change.*
- unsigned int [GetPlanarConfiguration](#) () const
- void [SetPlanarConfiguration](#) (unsigned int pc)  
*Set/Get requested PlanarConfiguration.*

## 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

### 10.145.1 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

### 10.145.2 Constructor & Destructor Documentation

#### 10.145.2.1 ImageChangePlanarConfiguration()

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

#### 10.145.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ( ) [inline]
```

### 10.145.3 Member Function Documentation

#### 10.145.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ( )
```

Change.

### 10.145.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]
```

### 10.145.3.3 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

### 10.145.3.4 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

### 10.145.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

Set/Get requested PlanarConfiguration.

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

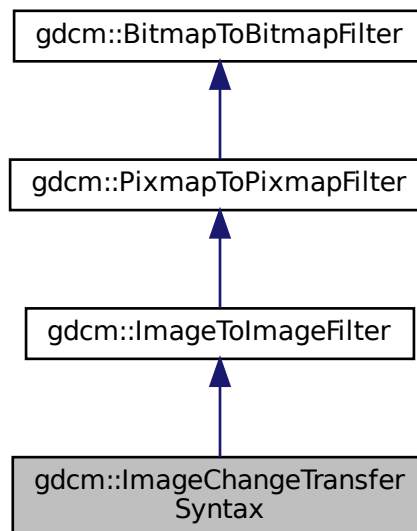
- [gdcmImageChangePlanarConfiguration.h](#)

## 10.146 gdcm::ImageChangeTransferSyntax Class Reference

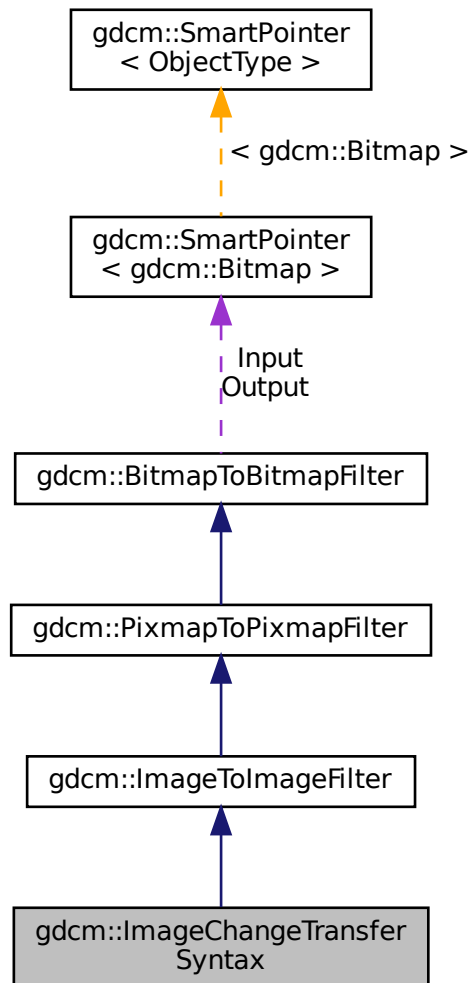
[ImageChangeTransferSyntax](#) class.

```
#include <gdcmImageChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



## Public Member Functions

- [ImageChangeTransferSyntax](#) ()
- [~ImageChangeTransferSyntax](#) ()
- [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)

## Protected Member Functions

- bool [TryJPEG2000Codec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGCodec](#) (const [DataElement](#) &pixelde, [Bitmap](#) const &input, [Bitmap](#) &output)
- bool [TryJPEGLSCodec](#) (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

### 10.146.1 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:

[CompressImage.cxx](#).

### 10.146.2 Constructor & Destructor Documentation

#### 10.146.2.1 ImageChangeTransferSyntax()

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

#### 10.146.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax ( ) [inline]
```



### 10.146.3 Member Function Documentation

#### 10.146.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ( )
```

Change.

Examples:

[CompressImage.cxx](#).

#### 10.146.3.2 GetTransferSyntax()

```
const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]
```

Get Transfer Syntax.

#### 10.146.3.3 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

#### 10.146.3.4 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

#### 10.146.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
    const TransferSyntax & ts ) [inline]
```

Set target Transfer Syntax.

Examples:

[CompressImage.cxx](#).

#### 10.146.3.6 SetUserCodec()

```
void gdcm::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 )`

#### 10.146.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

#### 10.146.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

## 10.146.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

## 10.146.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
    const DataElement & pixelde,
    Bitmap const & input,
    Bitmap & output ) [protected]
```

## 10.146.3.11 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:

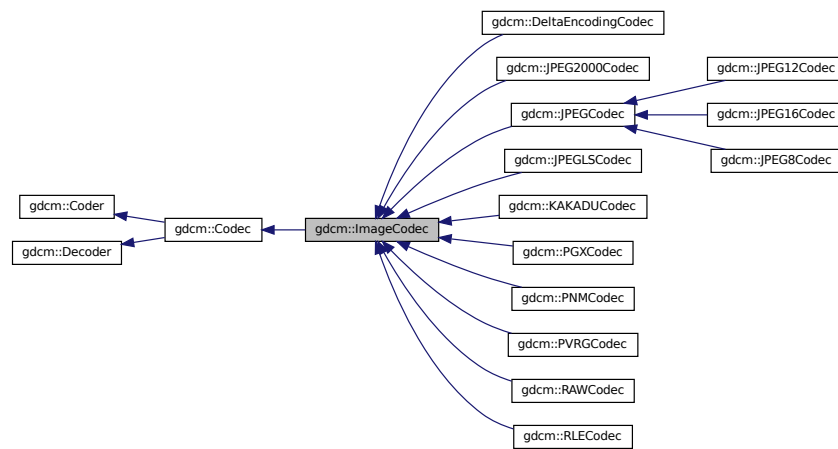
- [gdcmImageChangeTransferSyntax.h](#)

## 10.147 gdcm::ImageCodec Class Reference

[ImageCodec](#).

```
#include <gdcmImageCodec.h>
```

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for `gdcm::ImageCodec`:



## Public Member Functions

- [ImageCodec](#) ()
- [~ImageCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const  
*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)  
*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 unsigned int d[3])
- void [SetDimensions](#) (const std::vector< unsigned int > &d)
- 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)

## 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)
- 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)
- 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 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](#)

### 10.147.1 Detailed Description

[ImageCodec](#).

#### Note

Main codec, this is a central place for all implementation

## 10.147.2 Member Typedef Documentation

### 10.147.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcM::ImageCodec::LUTPtr [protected]
```

## 10.147.3 Constructor & Destructor Documentation

### 10.147.3.1 ImageCodec()

```
gdcM::ImageCodec::ImageCodec ( )
```

### 10.147.3.2 ~ImageCodec()

```
gdcM::ImageCodec::~~ImageCodec ( )
```

## 10.147.4 Member Function Documentation

### 10.147.4.1 AppendFrameEncode()

```
virtual bool gdcM::ImageCodec::AppendFrameEncode (  
    std::ostream & out,  
    const char * data,  
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcM::JPEGCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEG2000Codec](#), and [gdcM::RLECodec](#).

#### 10.147.4.2 AppendRowEncode()

```
virtual bool gdcmm::ImageCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::JPEG2000Codec](#), and [gdcmm::RLECodec](#).

#### 10.147.4.3 CanCode()

```
bool gdcmm::ImageCodec::CanCode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcmm::Coder](#).

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::PVRGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::PNMCodec](#), [gdcmm::PGXCodec](#), [gdcmm::KAKADUCodec](#), and [gdcmm::RAWCodec](#).

#### 10.147.4.4 CanDecode()

```
bool gdcmm::ImageCodec::CanDecode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcmm::Decoder](#).

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::PVRGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::PNMCodec](#), [gdcmm::RAWCodec](#), [gdcmm::PGXCodec](#), and [gdcmm::KAKADUCodec](#).

#### 10.147.4.5 CleanupUnusedBits()

```
bool gdcmm::ImageCodec::CleanupUnusedBits (
    char * data,
    size_t datalen )
```

#### 10.147.4.6 Clone()

```
virtual ImageCodec* gdcM::ImageCodec::Clone ( ) const [pure virtual]
```

Implemented in [gdcM::JPEGCodec](#), [gdcM::RLECodec](#), [gdcM::JPEGLSCodec](#), [gdcM::PVRGCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::PNMCodec](#), [gdcM::RAWCodec](#), [gdcM::KAKADUCodec](#), and [gdcM::PGXCodec](#).

#### 10.147.4.7 Decode()

```
bool gdcM::ImageCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcM::Decoder](#).

Reimplemented in [gdcM::JPEGCodec](#), [gdcM::RLECodec](#), [gdcM::PVRGCodec](#), [gdcM::JPEGLSCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::KAKADUCodec](#), and [gdcM::RAWCodec](#).

#### 10.147.4.8 DecodeByStreams()

```
bool gdcM::ImageCodec::DecodeByStreams (
    std::istream & is_,
    std::ostream & os ) [protected], [virtual]
```

Reimplemented from [gdcM::Decoder](#).

Reimplemented in [gdcM::JPEGCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::RLECodec](#), [gdcM::RAWCodec](#), [gdcM::JPEG12Codec](#), [gdcM::JPEG16Codec](#), and [gdcM::JPEG8Codec](#).

#### 10.147.4.9 DoByteSwap()

```
bool gdcM::ImageCodec::DoByteSwap (
    std::istream & is_,
    std::ostream & os ) [protected]
```



#### 10.147.4.10 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.11 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.12 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.13 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.14 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.15 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
    std::istream & is_,
    std::ostream & os ) [protected]
```

#### 10.147.4.16 GetDimensions()

```
const unsigned int* gdcm::ImageCodec::GetDimensions ( ) const [inline]
```

References `gdcm::terminal::dim`.

#### 10.147.4.17 GetHeaderInfo()

```
virtual bool gdcm::ImageCodec::GetHeaderInfo (
    std::istream & is_,
    TransferSyntax & ts ) [virtual]
```

Reimplemented in `gdcm::JPEGCodec`, `gdcm::RLECodec`, `gdcm::JPEGLSCodec`, `gdcm::JPEG2000Codec`, `gdcm::PNGCodec`, `gdcm::JPEG12Codec`, `gdcm::JPEG16Codec`, `gdcm::JPEG8Codec`, `gdcm::RAWCodec`, and `gdcm::PGXCodec`.

#### 10.147.4.18 GetLossyFlag()

```
bool gdcm::ImageCodec::GetLossyFlag ( ) const
```

#### 10.147.4.19 GetLUT()

```
const LookupTable& gdcm::ImageCodec::GetLUT ( ) const [inline]
```

#### 10.147.4.20 GetNeedByteSwap()

```
bool gdcm::ImageCodec::GetNeedByteSwap ( ) const [inline]
```

#### 10.147.4.21 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const
```

#### 10.147.4.22 GetPhotometricInterpretation()

```
const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const
```

#### 10.147.4.23 GetPixelFormat() [1/2]

```
PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) [inline]
```

Examples:

[GetJPEGSamplePrecision.cxx](#).

#### 10.147.4.24 GetPixelFormat() [2/2]

```
const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]
```

#### 10.147.4.25 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]
```

#### 10.147.4.26 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), and [gdcm::RLECodec](#).

#### 10.147.4.27 IsLossy()

```
bool gdcm::ImageCodec::IsLossy ( ) const
```

**10.147.4.28 IsRowEncoder()**

```
virtual bool gdcm::ImageCodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), and [gdcm::RLECodec](#).

**10.147.4.29 IsValid()**

```
virtual bool gdcm::ImageCodec::IsValid (
    PhotometricInterpretation const & pi ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#).

**10.147.4.30 SetDimensions()** [1/2]

```
void gdcm::ImageCodec::SetDimensions (
    const unsigned int d[3] )
```

**Examples:**

[ExtractIconFromFile.cxx](#).

**10.147.4.31 SetDimensions()** [2/2]

```
void gdcm::ImageCodec::SetDimensions (
    const std::vector< unsigned int > & d )
```

**10.147.4.32 SetLossyFlag()**

```
void gdcm::ImageCodec::SetLossyFlag (
    bool l )
```

#### 10.147.4.33 SetLUT()

```
void gdcm::ImageCodec::SetLUT (
    LookupTable const & lut ) [inline]
```

##### Examples:

[ExtractIconFromFile.cxx](#).

#### 10.147.4.34 SetNeedByteSwap()

```
void gdcm::ImageCodec::SetNeedByteSwap (
    bool b ) [inline]
```

#### 10.147.4.35 SetNeedOverlayCleanup()

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
    bool b ) [inline]
```

#### 10.147.4.36 SetNumberOfDimensions()

```
void gdcm::ImageCodec::SetNumberOfDimensions (
    unsigned int dim )
```

#### 10.147.4.37 SetPhotometricInterpretation()

```
void gdcm::ImageCodec::SetPhotometricInterpretation (
    PhotometricInterpretation const & pi )
```

##### Examples:

[ExtractIconFromFile.cxx](#).

#### 10.147.4.38 SetPixelFormat()

```
virtual void gdcm::ImageCodec::SetPixelFormat (
    PixelFormat const & pf ) [inline], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#).

Examples:

[ExtractIconFromFile.cxx](#).

#### 10.147.4.39 SetPlanarConfiguration()

```
void gdcm::ImageCodec::SetPlanarConfiguration (
    unsigned int pc ) [inline]
```

#### 10.147.4.40 StartEncode()

```
virtual bool gdcm::ImageCodec::StartEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), and [gdcm::RLECodec](#).

#### 10.147.4.41 StopEncode()

```
virtual bool gdcm::ImageCodec::StopEncode (
    std::ostream & os ) [protected], [virtual]
```

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::JPEGLSCodec](#), [gdcm::JPEG2000Codec](#), and [gdcm::RLECodec](#).

### 10.147.5 Friends And Related Function Documentation

### 10.147.5.1 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

### 10.147.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation [friend]
```

## 10.147.6 Member Data Documentation

### 10.147.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3] [protected]
```

### 10.147.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag [protected]
```

### 10.147.6.3 LUT

```
LUTPtr gdcm::ImageCodec::LUT [protected]
```

### 10.147.6.4 NeedByteSwap

```
bool gdcm::ImageCodec::NeedByteSwap [protected]
```

#### 10.147.6.5 NeedOverlayCleanup

```
bool gdcm::ImageCodec::NeedOverlayCleanup [protected]
```

#### 10.147.6.6 NumberOfDimensions

```
unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]
```

#### 10.147.6.7 PF

```
PixelFormat gdcm::ImageCodec::PF [protected]
```

#### 10.147.6.8 PI

```
PhotometricInterpretation gdcm::ImageCodec::PI [protected]
```

#### 10.147.6.9 PlanarConfiguration

```
unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]
```

#### 10.147.6.10 RequestPaddedCompositePixelCode

```
bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]
```

#### 10.147.6.11 RequestPlanarConfiguration

```
bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]
```

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

- [gdcmImageCodec.h](#)



## 10.148 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)

### 10.148.1 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

### 10.148.2 Constructor & Destructor Documentation

#### 10.148.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ( )
```

#### 10.148.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~~ImageConverter ( )
```

### 10.148.3 Member Function Documentation

### 10.148.3.1 Convert()

```
void gdcM::ImageConverter::Convert ( )
```

### 10.148.3.2 GetOutput()

```
const Image& gdcM::ImageConverter::GetOutput ( ) const
```

### 10.148.3.3 SetInput()

```
void gdcM::ImageConverter::SetInput (
    Image const & input )
```

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

- [gdcMImageConverter.h](#)

## 10.149 gdcM::ImageFragmentSplitter Class Reference

[ImageFragmentSplitter](#) class.

```
#include <gdcMImageFragmentSplitter.h>
```

Inheritance diagram for gdcM::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



## Public Member Functions

- `ImageFragmentSplitter ()`
- `~ImageFragmentSplitter ()`
- `unsigned int GetFragmentSizeMax () const`
- `void SetForce (bool f)`
- `void SetFragmentSizeMax (unsigned int fragsize)`  
*FragmentSizeMax needs to be an even number.*
- `bool Split ()`  
*Split.*

## Additional Inherited Members

### 10.149.1 Detailed Description

[ImageFragmentSplitter](#) class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

### 10.149.2 Constructor & Destructor Documentation

#### 10.149.2.1 ImageFragmentSplitter()

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

#### 10.149.2.2 ~ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ( ) [inline]
```

### 10.149.3 Member Function Documentation

#### 10.149.3.1 GetFragmentSizeMax()

```
unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]
```

#### 10.149.3.2 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 !

## 10.149.3.3 SetFragmentSizeMax()

```
void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (
    unsigned int fragsize )
```

FragmentSizeMax needs to be an even number.

## 10.149.3.4 Split()

```
bool gdcm::ImageFragmentSplitter::Split ( )
```

Split.

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

- [gdcmImageFragmentSplitter.h](#)

## 10.150 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 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 [SetSpacingValue](#) ([DataSet](#) &ds, const std::vector< double > &spacing)

## Static Protected Member Functions

- static [Tag](#) [GetSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)
- static [Tag](#) [GetZSpacingTagFromMediaStorage](#) ([MediaStorage](#) const &ms)

### 10.150.1 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.

### 10.150.2 Member Function Documentation

#### 10.150.2.1 ComputeMediaStorageFromModality()

```
static MediaStorage gdcmm::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](#).

### 10.150.2.2 ComputeSpacingFromImagePositionPatient()

```
static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
    const std::vector< double > & imageposition,
    std::vector< double > & spacing ) [static]
```

DO NOT USE.

### 10.150.2.3 GetDimensionsValue()

```
static 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:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.150.2.4 GetDirectionCosinesFromDataSet()

```
static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
    DataSet const & ds,
    std::vector< double > & dircos ) [static]
```

### 10.150.2.5 GetDirectionCosinesValue()

```
static 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

### 10.150.2.6 GetForcePixelSpacing()

```
static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]
```

#### 10.150.2.7 GetForceRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]
```

#### 10.150.2.8 GetLUT()

```
static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (
    File const & f ) [static]
```

returns the lookup table of an image file

#### 10.150.2.9 GetOriginValue()

```
static std::vector<double> gdcm::ImageHelper::GetOriginValue (
    File const & f ) [static]
```

Set/Get Origin (IPP) from/to a file.

#### 10.150.2.10 GetPhotometricInterpretationValue()

```
static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
    File const & f ) [static]
```

#### 10.150.2.11 GetPixelFormatValue()

```
static 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)

#### 10.150.2.12 GetPlanarConfigurationValue()

```
static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
    const File & f ) [static]
```



#### 10.150.2.13 GetPMSRescaleInterceptSlope()

```
static bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ( ) [static]
```

#### 10.150.2.14 GetPointerFromElement()

```
static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (
    Tag const & tag,
    File const & f ) [static]
```

#### 10.150.2.15 GetRealWorldValueMappingContent()

```
static bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
    File const & f,
    RealWorldValueMappingContent & rwvmc ) [static]
```

#### 10.150.2.16 GetRescaleInterceptSlopeValue()

```
static 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

#### 10.150.2.17 GetSpacingTagFromMediaStorage()

```
static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

#### 10.150.2.18 GetSpacingValue()

```
static std::vector<double> gdc::ImageHelper::GetSpacingValue (
    File const & f ) [static]
```

Set/Get [Spacing](#) from/to a [File](#).

#### 10.150.2.19 GetZSpacingTagFromMediaStorage()

```
static Tag gdc::ImageHelper::GetZSpacingTagFromMediaStorage (
    MediaStorage const & ms ) [static], [protected]
```

#### 10.150.2.20 SetDimensionsValue()

```
static void gdc::ImageHelper::SetDimensionsValue (
    File & f,
    const Pixmap & img ) [static]
```

#### 10.150.2.21 SetDirectionCosinesValue()

```
static void gdc::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.

#### 10.150.2.22 SetForcePixelSpacing()

```
static void gdc::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](#)

#### 10.150.2.23 SetForceRescaleInterceptSlope()

```
static 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.

#### 10.150.2.24 SetOriginValue()

```
static void gdcm::ImageHelper::SetOriginValue (
    DataSet & ds,
    const Image & img ) [static]
```

#### 10.150.2.25 SetPMSRescaleInterceptSlope()

```
static 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 ForceRescaleInterceptSlope is set to true GDCM will only read those private attribute but never write them out.

#### 10.150.2.26 SetRescaleInterceptSlopeValue()

```
static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
    File & f,
    const Image & img ) [static]
```

#### 10.150.2.27 SetSpacingValue()

```
static void gdcm::ImageHelper::SetSpacingValue (
    DataSet & ds,
    const std::vector< double > & spacing ) [static]
```

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

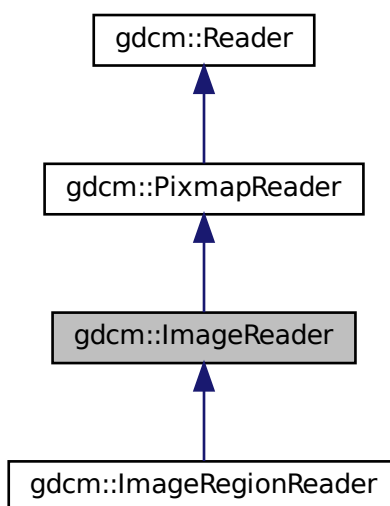
- [gdcmImageHelper.h](#)

## 10.151 gdcm::ImageReader Class Reference

[ImageReader.](#)

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



## Public Member Functions

- [ImageReader](#) ()
- virtual [~ImageReader](#) ()
- const [Image](#) & [GetImage](#) () const  
*Return the read image.*
- [Image](#) & [GetImage](#) ()
- virtual bool [Read](#) ()

## Protected Member Functions

- bool [ReadACRNEMAIImage](#) ()
- bool [ReadImage](#) ([MediaStorage](#) const &ms)

## Additional Inherited Members

### 10.151.1 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:

[CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 10.151.2 Constructor & Destructor Documentation

#### 10.151.2.1 ImageReader()

```
gdcm::ImageReader::ImageReader ( )
```

#### 10.151.2.2 ~ImageReader()

```
virtual gdcm::ImageReader::~~ImageReader ( ) [virtual]
```

### 10.151.3 Member Function Documentation

### 10.151.3.1 GetImage() [1/2]

```
const Image& gdcm::ImageReader::GetImage ( ) const
```

Return the read image.

#### Examples:

[CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

### 10.151.3.2 GetImage() [2/2]

```
Image& gdcm::ImageReader::GetImage ( )
```

### 10.151.3.3 Read()

```
virtual bool gdcm::ImageReader::Read ( ) [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::PixmapReader](#).

Reimplemented in [gdcm::ImageRegionReader](#).

#### Examples:

[CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 10.151.3.4 ReadACRNEMAImage()

```
bool gdcm::ImageReader::ReadACRNEMAImage ( ) [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

### 10.151.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
    MediaStorage const & ms ) [protected], [virtual]
```

Reimplemented from [gdcm::PixmapReader](#).

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

- [gdcmImageReader.h](#)

## 10.152 gdcm::ImageRegionReader Class Reference

[ImageRegionReader](#).

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:





Collaboration diagram for gdcm::ImageRegionReader:



## Public Member Functions

- [ImageRegionReader](#) ()
- [~ImageRegionReader](#) ()
- [size\\_t ComputeBufferLength](#) () const
- [Region](#) const & [GetRegion](#) () const
- [bool ReadInformation](#) ()
- [bool ReadIntoBuffer](#) (char \*inreadbuffer, size\_t buflen)
- [void SetRegion](#) ([Region](#) const &region)

*Set/Get [Region](#) to be read.*

## Protected Member Functions

- bool [Read](#) ()

*To prevent user from calling super class [Read\(\)](#) function.*

## Additional Inherited Members

### 10.152.1 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](http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_8.2.↵html#para_4bcb841e-c6bf-4e26-82a5-3fad3c942da0)

See also

[ImageReader](#)

Examples:

[TemplateEmptyImage.cxx](#).

### 10.152.2 Constructor & Destructor Documentation

#### 10.152.2.1 ImageRegionReader()

```
gdcm::ImageRegionReader::ImageRegionReader ( )
```

#### 10.152.2.2 ~ImageRegionReader()

```
gdcm::ImageRegionReader::~~ImageRegionReader ( )
```

### 10.152.3 Member Function Documentation

### 10.152.3.1 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

### 10.152.3.2 GetRegion()

```
Region const& gdcm::ImageRegionReader::GetRegion ( ) const
```

### 10.152.3.3 Read()

```
bool gdcm::ImageRegionReader::Read ( ) [protected], [virtual]
```

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

### 10.152.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ( )
```

Read meta information (not Pixel Data) from the DICOM file.

#### Returns

false upon error

#### Examples:

[TemplateEmptyImage.cxx](#).

### 10.152.3.5 ReadIntoBuffer()

```
bool gdcM::ImageRegionReader::ReadIntoBuffer (
    char * inreadbuffer,
    size_t buflen )
```

Read into buffer:

#### Returns

false upon error

### 10.152.3.6 SetRegion()

```
void gdcM::ImageRegionReader::SetRegion (
    Region const & region )
```

Set/Get [Region](#) to be read.

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

- [gdcMImageRegionReader.h](#)

## 10.153 gdcM::ImageToImageFilter Class Reference

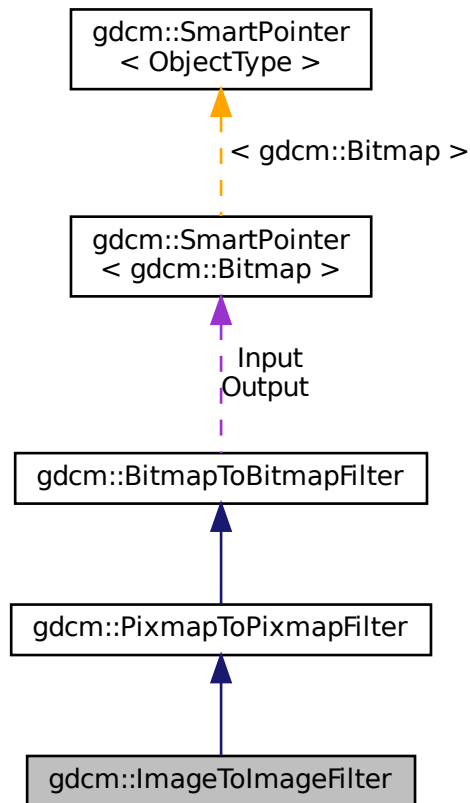
[ImageToImageFilter](#) class.

```
#include <gdcMImageToImageFilter.h>
```

Inheritance diagram for gdcM::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



### Public Member Functions

- [ImageToImageFilter](#) ()
- [~ImageToImageFilter](#) ()
- [Image](#) & [GetInput](#) ()
- const [Image](#) & [GetOutput](#) () const  
*Get Output image.*

### Additional Inherited Members

#### 10.153.1 Detailed Description

[ImageToImageFilter](#) class.

Super class for all filter taking an image and producing an output image

### 10.153.2 Constructor & Destructor Documentation

#### 10.153.2.1 ImageToImageFilter()

```
gdcm::ImageToImageFilter::ImageToImageFilter ( )
```

#### 10.153.2.2 ~ImageToImageFilter()

```
gdcm::ImageToImageFilter::~~ImageToImageFilter ( ) [inline]
```

### 10.153.3 Member Function Documentation

#### 10.153.3.1 GetInput()

```
Image& gdcm::ImageToImageFilter::GetInput ( )
```

#### 10.153.3.2 GetOutput()

```
const Image& gdcm::ImageToImageFilter::GetOutput ( ) const
```

Get Output image.

Examples:

[CompressImage.cxx](#).

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

- [gdcmImageToImageFilter.h](#)

## 10.154 gdcm::ImageWriter Class Reference

[ImageWriter.](#)

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for `gdcm::ImageWriter`:



## Public Member Functions

- [ImageWriter](#) ()
- [~ImageWriter](#) ()
- [MediaStorage ComputeTargetMediaStorage](#) ()
- `const Image & GetImage () const`
- `Image & GetImage ()`
- `bool Write ()`

*Write.*



## Additional Inherited Members

### 10.154.1 Detailed Description

[ImageWriter](#).

Examples:

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [MergeTwoFiles.cxx](#), and [TemplateEmptyImage.cxx](#).

### 10.154.2 Constructor & Destructor Documentation

#### 10.154.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ( )
```

#### 10.154.2.2 ~ImageWriter()

```
gdcm::ImageWriter::~ImageWriter ( )
```

### 10.154.3 Member Function Documentation

#### 10.154.3.1 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](#).

**10.154.3.2 GetImage()** [1/2]

```
const Image& gdcm::ImageWriter::GetImage ( ) const [inline], [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](#).

**10.154.3.3 GetImage()** [2/2]

```
Image& gdcm::ImageWriter::GetImage ( ) [inline], [virtual]
```

Reimplemented from [gdcm::PixmapWriter](#).

**10.154.3.4 Write()**

```
bool gdcm::ImageWriter::Write ( ) [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

**Examples:**

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [MergeTwoFiles.cxx](#), and [TemplateEmptyImage.cxx](#).

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

- [gdcmImageWriter.h](#)

**10.155 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

### 10.155.1 Detailed Description

[ImplementationClassUIDSub](#).

PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.155.2 Constructor & Destructor Documentation

#### 10.155.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )
```

### 10.155.3 Member Function Documentation

#### 10.155.3.1 Print()

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

#### 10.155.3.2 Read()

```
std::istream& gdcm::network::ImplementationClassUIDSub::Read (
    std::istream & is )
```

### 10.155.3.3 Size()

```
size_t gdcn::network::ImplementationClassUIDSub::Size ( ) const
```

### 10.155.3.4 Write()

```
const std::ostream& gdcn::network::ImplementationClassUIDSub::Write (
    std::ostream & os ) const
```

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

- [gdcnImplementationClassUIDSub.h](#)

## 10.156 gdcn::network::ImplementationUIDSub Class Reference

[ImplementationUIDSub](#).

```
#include <gdcnImplementationUIDSub.h>
```

### Public Member Functions

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.156.1 Detailed Description

[ImplementationUIDSub](#).

[Table](#) D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

### 10.156.2 Constructor & Destructor Documentation

#### 10.156.2.1 ImplementationUIDSub()

```
gdcn::network::ImplementationUIDSub::ImplementationUIDSub ( )
```

### 10.156.3 Member Function Documentation

#### 10.156.3.1 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](#)

## 10.157 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

### 10.157.1 Detailed Description

[ImplementationVersionNameSub.](#)

[Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.157.2 Constructor & Destructor Documentation

#### 10.157.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )
```

### 10.157.3 Member Function Documentation

#### 10.157.3.1 Print()

```
void gdcM::network::ImplementationVersionNameSub::Print (
    std::ostream & os ) const
```

#### 10.157.3.2 Read()

```
std::istream& gdcM::network::ImplementationVersionNameSub::Read (
    std::istream & is )
```

#### 10.157.3.3 Size()

```
size_t gdcM::network::ImplementationVersionNameSub::Size ( ) const
```

#### 10.157.3.4 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](#)

## 10.158 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`

## Additional Inherited Members

### 10.158.1 Detailed Description

Class to represent an *Implicit* [VR](#) Data [Element](#).

#### Note

bla

#### Examples:

[ReadExplicitLengthSQIVR.cxx](#).

### 10.158.2 Member Function Documentation

#### 10.158.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength ( ) const
```

#### 10.158.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::Read (
    std::istream & is )
```



### 10.158.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadPreValue (
    std::istream & is )
```

### 10.158.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

### 10.158.2.5 ReadValueWithLength()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadValueWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true )
```

### 10.158.2.6 ReadWithLength()

```
template<typename TSwap >
std::istream& gdcm::ImplicitDataElement::ReadWithLength (
    std::istream & is,
    VL & length,
    bool readvalues = true )
```

### 10.158.2.7 Write()

```
template<typename TSwap >
const std::ostream& gdcm::ImplicitDataElement::Write (
    std::ostream & os ) const
```

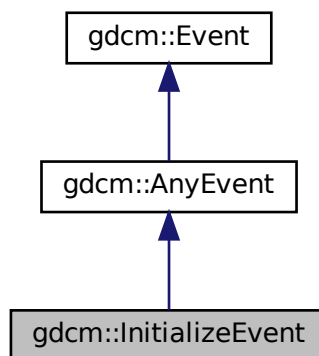
The documentation for this class was generated from the following file:

- [gdcmImplicitDataElement.h](#)

## 10.159 gdcm::InitializeEvent Class Reference

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

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.160 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](#) ()
- 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)

#### 10.160.1 Detailed Description

Class for representing a [IOD](#).

#### Note

bla

#### See also

[Dict](#)

#### Examples:

[TraverseModules.cxx](#).

#### 10.160.2 Member Typedef Documentation

### 10.160.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

### 10.160.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

## 10.160.3 Constructor & Destructor Documentation

### 10.160.3.1 IOD()

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

References `gdcm::operator<<()`.

## 10.160.4 Member Function Documentation

### 10.160.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
    const IODEntry & iode ) [inline]
```

### 10.160.4.2 Clear()

```
void gdcm::IOD::Clear ( ) [inline]
```

#### 10.160.4.3 GetIODEntry()

```
const IODEntry& gdcm::IOD::GetIODEntry (
    SizeType idx ) const [inline]
```

Examples:

[TraverseModules.cxx](#).

#### 10.160.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs ( ) const [inline]
```

Examples:

[TraverseModules.cxx](#).

#### 10.160.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (
    const Defs & defs,
    const Tag & tag ) const
```

### 10.160.5 Friends And Related Function Documentation

#### 10.160.5.1 operator<<

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

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

- [gdcmIOD.h](#)

## 10.161 gdcm::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmIODEntry.h>
```

### Public Member Functions

- [IODEntry](#) (const char \*name="", const char \*ref="", const char \*usag="")
- 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 \*usag)

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [IODEntry](#) &\_val)

### 10.161.1 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](#).

## 10.161.2 Constructor & Destructor Documentation

### 10.161.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
    const char * name = "",
    const char * ref = "",
    const char * usag = "" ) [inline]
```

References `gdcm::operator<<()`.

## 10.161.3 Member Function Documentation

### 10.161.3.1 GetIE()

```
const char* gdcm::IODEntry::GetIE ( ) const [inline]
```

### 10.161.3.2 GetName()

```
const char* gdcm::IODEntry::GetName ( ) const [inline]
```

### 10.161.3.3 GetRef()

```
const char* gdcm::IODEntry::GetRef ( ) const [inline]
```

#### Examples:

[TraverseModules.cxx](#).

### 10.161.3.4 GetUsage()

```
const char* gdcm::IODEntry::GetUsage ( ) const [inline]
```

#### 10.161.3.5 GetUsageType()

```
Usage::UsageType gdcM::IODEntry::GetUsageType ( ) const
```

#### 10.161.3.6 SetIE()

```
void gdcM::IODEntry::SetIE (
    const char * ie ) [inline]
```

#### 10.161.3.7 SetName()

```
void gdcM::IODEntry::SetName (
    const char * name ) [inline]
```

#### 10.161.3.8 SetRef()

```
void gdcM::IODEntry::SetRef (
    const char * ref ) [inline]
```

#### 10.161.3.9 SetUsage()

```
void gdcM::IODEntry::SetUsage (
    const char * usag ) [inline]
```

### 10.161.4 Friends And Related Function Documentation

#### 10.161.4.1 operator<<

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

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

- [gdcMIODEntry.h](#)



## 10.162 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](#) ()
- 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)

#### 10.162.1 Detailed Description

Class for representing a [IODs](#).

#### Note

bla

#### See also

[IOD](#)

#### Examples:

[TraverseModules.cxx](#).

#### 10.162.2 Member Typedef Documentation

### 10.162.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdc::IODs::IODMapType
```

### 10.162.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const_iterator gdc::IODs::IODMapTypeConstIterator
```

### 10.162.2.3 IODName

```
typedef std::string gdc::IODs::IODName
```

## 10.162.3 Constructor & Destructor Documentation

### 10.162.3.1 IODs()

```
gdc::IODs::IODs ( ) [inline]
```

References `gdc::operator<<()`.

## 10.162.4 Member Function Documentation

### 10.162.4.1 AddIOD()

```
void gdc::IODs::AddIOD (
    const char * name,
    const IOD & module ) [inline]
```

#### 10.162.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]
```

Examples:

[TraverseModules.cxx](#).

#### 10.162.4.3 Clear()

```
void gdcm::IODs::Clear ( ) [inline]
```

#### 10.162.4.4 End()

```
IODMapTypeConstIterator gdcm::IODs::End ( ) const [inline]
```

Examples:

[TraverseModules.cxx](#).

#### 10.162.4.5 GetIOD()

```
const IOD& gdcm::IODs::GetIOD (
    const char * name ) const [inline]
```

### 10.162.5 Friends And Related Function Documentation

#### 10.162.5.1 operator<<

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

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

- [gdcmIODs.h](#)



## Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

## Additional Inherited Members

### 10.163.1 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](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](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

### 10.163.2 Constructor & Destructor Documentation

#### 10.163.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ( )
```

### 10.163.3 Member Function Documentation

#### 10.163.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]
```

#### 10.163.3.2 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](#).

#### 10.163.3.3 GetZSpacingTolerance()

```
double gdcm::IPPSorter::GetZSpacingTolerance ( ) const [inline]
```

#### 10.163.3.4 SetComputeZSpacing()

```
void gdcm::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](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

## 10.163.3.5 SetDirectionCosinesTolerance()

```
void gdcm::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).

## 10.163.3.6 SetDropDuplicatePositions()

```
void gdcm::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.

## 10.163.3.7 SetZSpacingTolerance()

```
void gdcm::IPPSorter::SetZSpacingTolerance (
    double tol ) [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the serie, a human can determine that this is ok and change the tolerance from its default value: 1e-6

## Examples:

[Compute3DSpacing.cxx](#), [gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

## 10.163.3.8 Sort()

```
virtual bool gdcm::IPPSorter::Sort (
    std::vector< std::string > const & filenames ) [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 achived. 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](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

### 10.163.4 Member Data Documentation

#### 10.163.4.1 ComputeZSpacing

```
bool gdc::IPPSorter::ComputeZSpacing [protected]
```

#### 10.163.4.2 DirCosTolerance

```
double gdc::IPPSorter::DirCosTolerance [protected]
```

#### 10.163.4.3 DropDuplicatePositions

```
bool gdc::IPPSorter::DropDuplicatePositions [protected]
```

#### 10.163.4.4 ZSpacing

```
double gdc::IPPSorter::ZSpacing [protected]
```

#### 10.163.4.5 ZTolerance

```
double gdc::IPPSorter::ZTolerance [protected]
```

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

- [gdcmlPPSorter.h](#)



## 10.164 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
- const [DataSet](#) & [GetNestedDataSet](#) () const
- [DataSet](#) & [GetNestedDataSet](#) ()
- 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

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Item](#) &val)

## Additional Inherited Members

### 10.164.1 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 Standart 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](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

### 10.164.2 Constructor & Destructor Documentation

### 10.164.2.1 Item() [1/2]

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

References `gdcm::operator<<()`.

### 10.164.2.2 Item() [2/2]

```
gdcm::Item::Item (
    Item const & val ) [inline]
```

## 10.164.3 Member Function Documentation

### 10.164.3.1 Clear()

```
void gdcm::Item::Clear ( ) [inline]
```

Referenced by `gdcm::SequenceOfItems::Read()`.

### 10.164.3.2 FindDataElement()

```
bool gdcm::Item::FindDataElement (
    const Tag & t ) const [inline]
```

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

### 10.164.3.3 GetDataElement()

```
const DataElement& gdcm::Item::GetDataElement (
    const Tag & t ) const [inline]
```

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

**10.164.3.4 GetLength()**

```
template<typename TDE >
VL gdcM::Item::GetLength ( ) const
```

**10.164.3.5 GetNestedDataSet()** [1/2]

```
const DataSet& gdcM::Item::GetNestedDataSet ( ) const [inline]
```

**Examples:**

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

Referenced by `gdcM::SequenceOfItems::Read()`.

**10.164.3.6 GetNestedDataSet()** [2/2]

```
DataSet& gdcM::Item::GetNestedDataSet ( ) [inline]
```

**10.164.3.7 InsertDataElement()**

```
void gdcM::Item::InsertDataElement (
    const DataSet& de ) [inline]
```

**10.164.3.8 Read()**

```
template<typename TDE , typename TSwap >
std::istream& gdcM::Item::Read (
    std::istream & is ) [inline]
```

References `gdcM::DataSet::Clear()`, `gdcMDebugMacro`, `gdcMErrorMacro`, `gdcMWarningMacro`, and `gdcM::DataSet::IsEmpty()`.

Referenced by `gdcM::SequenceOfItems::Read()`.

### 10.164.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
    const DataSet & nested ) [inline]
```

### 10.164.3.10 Write()

```
template<typename TDE , typename TSwap >
const std::ostream& gdcm::Item::Write (
    std::ostream & os ) const [inline]
```

References `gdcmWarningMacro`, `gdcm::VL::GetLength()`, `gdcm::VL::Write()`, and `gdcm::Tag::Write()`.

## 10.164.4 Friends And Related Function Documentation

### 10.164.4.1 operator<<

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

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

- [gdcmItem.h](#)

## 10.165 gdcm::IterationEvent Class Reference

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

Inheritance diagram for `gdcm::IterationEvent`:



Collaboration diagram for `gdcm::IterationEvent`:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.166 gdcm::JPEG12Codec Class Reference

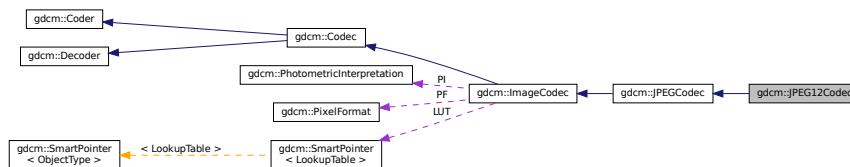
Class to do JPEG 12bits (lossy & lossless)

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



## Public Member Functions

- [JPEG12Codec](#) ()
- [~JPEG12Codec](#) ()
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os)

## Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen)
- bool [IsStateSuspension](#) () const

## Additional Inherited Members

### 10.166.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

#### Note

internal class

### 10.166.2 Constructor & Destructor Documentation

#### 10.166.2.1 JPEG12Codec()

```
gdcmm::JPEG12Codec::JPEG12Codec ( )
```

#### 10.166.2.2 ~JPEG12Codec()

```
gdcmm::JPEG12Codec::~~JPEG12Codec ( )
```

### 10.166.3 Member Function Documentation

#### 10.166.3.1 DecodeByStreams()

```
bool gdcmm::JPEG12Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcmm::ImageCodec](#).



### 10.166.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG12Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.166.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.166.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.166.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

- [gdcmJPEG12Codec.h](#)

## 10.167 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](#) ()
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os)

## Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen)
- bool [IsStateSuspension](#) () const

## Additional Inherited Members

### 10.167.1 Detailed Description

Class to do JPEG 16bits (lossless)

#### Note

internal class

### 10.167.2 Constructor & Destructor Documentation

#### 10.167.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ( )
```

#### 10.167.2.2 ~JPEG16Codec()

```
gdcm::JPEG16Codec::~~JPEG16Codec ( )
```

### 10.167.3 Member Function Documentation

### 10.167.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.167.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG16Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.167.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.167.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcm::Coder](#).

### 10.167.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

- [gdcmJPEG16Codec.h](#)

## 10.168 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](#) ()
- [bool CanCode](#) ([TransferSyntax](#) const &ts) const

*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const  
*Return whether this decoder support this transfer syntax (can decode it)*
- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*
- virtual bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- double [GetQuality](#) (unsigned int idx=0) const
- double [GetRate](#) (unsigned int idx=0) const
- void [SetNumberOfResolutions](#) (unsigned int nres)
- 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)

### Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

### Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

### Additional Inherited Members

#### 10.168.1 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](#)

## 10.168.2 Constructor & Destructor Documentation

### 10.168.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ( )
```

### 10.168.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~~JPEG2000Codec ( )
```

## 10.168.3 Member Function Documentation

### 10.168.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.5 Clone()

```
virtual ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.168.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).



### 10.168.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.9 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]
```

### 10.168.3.10 GetHeaderInfo()

```
virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

**10.168.3.11 GetQuality()**

```
double gdcM::JPEG2000Codec::GetQuality (
    unsigned int idx = 0 ) const
```

**10.168.3.12 GetRate()**

```
double gdcM::JPEG2000Codec::GetRate (
    unsigned int idx = 0 ) const
```

**10.168.3.13 IsFrameEncoder()**

```
bool gdcM::JPEG2000Codec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.168.3.14 IsRowEncoder()**

```
bool gdcM::JPEG2000Codec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

**10.168.3.15 SetNumberOfResolutions()**

```
void gdcM::JPEG2000Codec::SetNumberOfResolutions (
    unsigned int nres )
```

**10.168.3.16 SetQuality()**

```
void gdcM::JPEG2000Codec::SetQuality (
    unsigned int idx,
    double q )
```

### 10.168.3.17 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
    unsigned int idx,
    double rate )
```

### 10.168.3.18 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
    bool res )
```

### 10.168.3.19 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
    unsigned int tx,
    unsigned int ty )
```

### 10.168.3.20 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.168.3.21 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.168.4 Friends And Related Function Documentation

#### 10.168.4.1 Bitmap

```
friend class Bitmap [friend]
```

#### 10.168.4.2 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

- [gdcmJPEG2000Codec.h](#)

### 10.169 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](#) ()
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [InternalCode](#) (const char \*input, unsigned long len, std::ostream &os)

## Protected Member Functions

- virtual bool [EncodeBuffer](#) (std::ostream &os, const char \*data, size\_t datalen)
- bool [IsStateSuspension](#) () const

## Additional Inherited Members

### 10.169.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

#### Note

internal class

### 10.169.2 Constructor & Destructor Documentation

#### 10.169.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ( )
```

### 10.169.2.2 ~JPEG8Codec()

```
gdcm::JPEG8Codec::~~JPEG8Codec ( )
```

## 10.169.3 Member Function Documentation

### 10.169.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.169.3.2 EncodeBuffer()

```
virtual bool gdcm::JPEG8Codec::EncodeBuffer (
    std::ostream & os,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.169.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

### 10.169.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
    const char * input,
    unsigned long len,
    std::ostream & os ) [virtual]
```

Reimplemented from [gdcm::Coder](#).

## 10.169.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented from [gdcm::JPEGCodec](#).

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

- [gdcmJPEG8Codec.h](#)

## 10.170 gdcm::JPEGCodec Class Reference

JPEG codec.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



## Public Member Functions

- [JPEGCodec](#) ()
- [~JPEGCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const  
*Return whether this decoder support this transfer syntax (can decode it)*
- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)  
*Compress into JPEG.*
- void [ComputeOffsetTable](#) (bool b)  
*Compute the offset table:*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*
- virtual bool [EncodeBuffer](#) (std::ostream &out, const char \*inbuffer, size\_t inlen)
- virtual bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [GetLossless](#) () const
- double [GetQuality](#) () const
- void [SetLossless](#) (bool l)
- void [SetPixelFormat](#) ([PixelFormat](#) const &pf)
- void [SetQuality](#) (double q)

## Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- void [SetBitSample](#) (int bit)
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

## Protected Attributes

- int [BitSample](#)
- int [Quality](#)

## Friends

- class [ImageRegionReader](#)



## Additional Inherited Members

### 10.170.1 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/625e46919f208](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/75fdfccc65a62](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/2d525ef6a2f09](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09)
- [http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/6b93af410f8c92](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c92)

#### Examples:

[GetJPEGSamplePrecision.cxx](#).

### 10.170.2 Constructor & Destructor Documentation

#### 10.170.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ( )
```

#### 10.170.2.2 ~JPEGCodec()

```
gdcm::JPEGCodec::~~JPEGCodec ( )
```

### 10.170.3 Member Function Documentation

### 10.170.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.5 Clone()

```
virtual ImageCodec\* gdcm::JPEGCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.170.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
    DataElement const & in,
    DataElement & out ) [virtual]
```

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

### 10.170.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
    bool b )
```

Compute the offset table:

### 10.170.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.170.3.10 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]
```

### 10.170.3.11 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](#).

### 10.170.3.12 GetHeaderInfo()

```
virtual bool gdcm::JPEGCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

Examples:

[GetJPEGSamplePrecision.cxx](#).

### 10.170.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless ( ) const
```

#### 10.170.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality ( ) const
```

#### 10.170.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.170.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.170.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const [protected], [virtual]
```

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

#### 10.170.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
    PhotometricInterpretation const & pi ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.170.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
    int bit ) [protected]
```

**10.170.3.20 SetLossless()**

```
void gdcm::JPEGCodec::SetLossless (
    bool l )
```

**10.170.3.21 SetPixelFormat()**

```
void gdcm::JPEGCodec::SetPixelFormat (
    PixelFormat const & pf ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

**Examples:**

[GetJPEGSamplePrecision.cxx](#).

**10.170.3.22 SetQuality()**

```
void gdcm::JPEGCodec::SetQuality (
    double q )
```

**10.170.3.23 StartEncode()**

```
bool gdcm::JPEGCodec::StartEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

**10.170.3.24 StopEncode()**

```
bool gdcm::JPEGCodec::StopEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.170.4 Friends And Related Function Documentation

### 10.170.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

## 10.170.5 Member Data Documentation

### 10.170.5.1 BitSample

```
int gdcm::JPEGCodec::BitSample [protected]
```

### 10.170.5.2 Quality

```
int gdcm::JPEGCodec::Quality [protected]
```

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

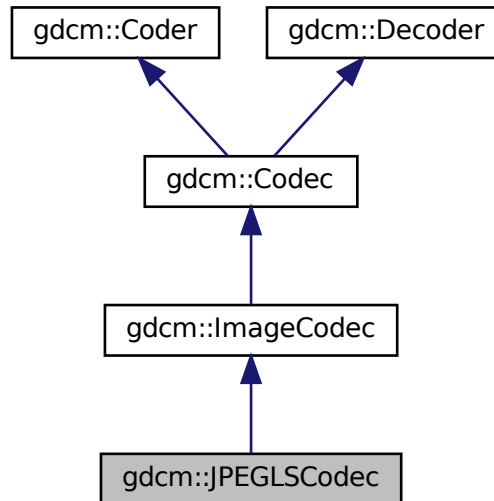
- [gdcmJPEGCodec.h](#)

## 10.171 gdcm::JPEGLSCodec Class Reference

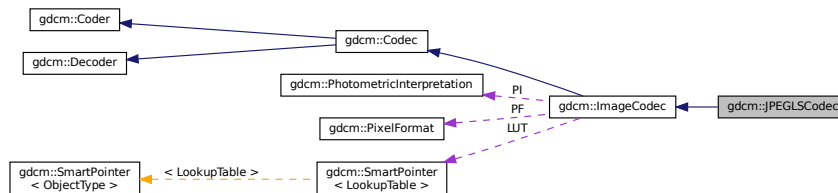
JPEG-LS.

```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for `gdcm::JPEGLSCodec`:



Collaboration diagram for `gdcm::JPEGLSCodec`:



## Public Member Functions

- `JPEGLSCodec` ()
- `~JPEGLSCodec` ()
- `bool CanCode (TransferSyntax const &ts) const`  
*Return whether this coder support this transfer syntax (can code it)*
- `bool CanDecode (TransferSyntax const &ts) const`  
*Return whether this decoder support this transfer syntax (can decode it)*
- `virtual ImageCodec * Clone () const`
- `bool Code (DataElement const &in, DataElement &out)`  
*Code.*



- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*
- 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)
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [GetLossless](#) () const
- void [SetBufferLength](#) (unsigned long l)
- void [SetLossless](#) (bool l)
- void [SetLossyError](#) (int error)  
*[0-3] generally*

### Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

### Friends

- class [ImageRegionReader](#)

### Additional Inherited Members

#### 10.171.1 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>

#### 10.171.2 Constructor & Destructor Documentation

### 10.171.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ( )
```

### 10.171.2.2 ~JPEGLSCodec()

```
gdcm::JPEGLSCodec::~~JPEGLSCodec ( )
```

## 10.171.3 Member Function Documentation

### 10.171.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.5 Clone()

```
virtual ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.171.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

#### 10.171.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.8 Decode() [2/2]

```
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 )
```

### 10.171.3.9 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]
```

### 10.171.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]
```

### 10.171.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless ( ) const
```

### 10.171.3.13 IsFrameEncoder()

```
bool gdcm::JPEGLSCodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.14 IsRowEncoder()

```
bool gdcm::JPEGLSCodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.3.15 SetBufferLength()

```
void gdcm::JPEGLSCodec::SetBufferLength (
    unsigned long l ) [inline]
```

### 10.171.3.16 SetLossless()

```
void gdcm::JPEGLSCodec::SetLossless (
    bool l )
```

### 10.171.3.17 SetLossyError()

```
void gdcm::JPEGLSCodec::SetLossyError (
    int error )
```

[0-3] generally

#### 10.171.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.171.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.171.4 Friends And Related Function Documentation

#### 10.171.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

- [gdcmJPEGLSCodec.h](#)

## 10.172 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)

### 10.172.1 Detailed Description

Examples:

[QIDO-RS.cxx](#).

### 10.172.2 Constructor & Destructor Documentation

#### 10.172.2.1 JSON()

```
gdcm::JSON::JSON ( )
```

#### 10.172.2.2 ~JSON()

```
gdcm::JSON::~~JSON ( )
```

### 10.172.3 Member Function Documentation

#### 10.172.3.1 Code()

```
bool gdcm::JSON::Code (
    DataSet const & in,
    std::ostream & os )
```

Examples:

[QIDO-RS.cxx](#).

#### 10.172.3.2 Decode()

```
bool gdcm::JSON::Decode (
    std::istream & is,
    DataSet & out )
```

Examples:

[QIDO-RS.cxx](#).

#### 10.172.3.3 GetPrettyPrint()

```
bool gdcM::JSON::GetPrettyPrint ( ) const
```

#### 10.172.3.4 PrettyPrintOff()

```
void gdcM::JSON::PrettyPrintOff ( )
```

#### 10.172.3.5 PrettyPrintOn()

```
void gdcM::JSON::PrettyPrintOn ( )
```

#### Examples:

[QIDO-RS.cxx](#).

#### 10.172.3.6 SetPrettyPrint()

```
void gdcM::JSON::SetPrettyPrint (
    bool onoff )
```

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

- [gdcMJSON.h](#)



## 10.173 gdcm::KAKADUCodec Class Reference

[KAKADUCodec](#).

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



### Public Member Functions

- [KAKADUCodec](#) ()
- [~KAKADUCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const

*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

*Return whether this decoder support this transfer syntax (can decode it)*

- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)

*Code.*

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

## Additional Inherited Members

### 10.173.1 Detailed Description

[KAKADUCodec](#).

### 10.173.2 Constructor & Destructor Documentation

#### 10.173.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ( )
```

#### 10.173.2.2 ~KAKADUCodec()

```
gdcm::KAKADUCodec::~~KAKADUCodec ( )
```

### 10.173.3 Member Function Documentation

#### 10.173.3.1 CanCode()

```
bool gdcm::KAKADUCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.173.3.2 CanDecode()

```
bool gdcm::KAKADUCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.173.3.3 Clone()

```
virtual ImageCodec* gdcm::KAKADUCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.173.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.173.3.5 Decode()

```
bool gdcm::KAKADUCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

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

- [gdcmKAKADUCodec.h](#)

## 10.174 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 [value\\_type](#) \*s)
- [LO](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- [LO](#) (const [Superclass](#) &s, [size\\_type](#) pos=0, [size\\_type](#) n=npos)
- bool [IsValid](#) () const

### 10.174.1 Detailed Description

[LO](#).

Note

TODO

### 10.174.2 Member Typedef Documentation

#### 10.174.2.1 `const_iterator`

```
typedef Superclass::const_iterator gdcm::LO::const\_iterator
```

#### 10.174.2.2 `const_reference`

```
typedef Superclass::const_reference gdcm::LO::const\_reference
```

#### 10.174.2.3 `const_reverse_iterator`

```
typedef Superclass::const_reverse_iterator gdcm::LO::const\_reverse\_iterator
```

#### 10.174.2.4 `difference_type`

```
typedef Superclass::difference_type gdcm::LO::difference\_type
```

**10.174.2.5 iterator**

```
typedef Superclass::iterator gdcM::LO::iterator
```

**10.174.2.6 pointer**

```
typedef Superclass::pointer gdcM::LO::pointer
```

**10.174.2.7 reference**

```
typedef Superclass::reference gdcM::LO::reference
```

**10.174.2.8 reverse\_iterator**

```
typedef Superclass::reverse_iterator gdcM::LO::reverse_iterator
```

**10.174.2.9 size\_type**

```
typedef Superclass::size_type gdcM::LO::size_type
```

**10.174.2.10 Superclass**

```
typedef String<'\\', 64> gdcM::LO::Superclass
```

**10.174.2.11 value\_type**

```
typedef Superclass::value_type gdcM::LO::value_type
```

### 10.174.3 Constructor & Destructor Documentation

#### 10.174.3.1 LO() [1/4]

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

#### 10.174.3.2 LO() [2/4]

```
gdcm::LO::LO (
    const value\_type * s ) [inline]
```

#### 10.174.3.3 LO() [3/4]

```
gdcm::LO::LO (
    const value\_type * s,
    size\_type n ) [inline]
```

#### 10.174.3.4 LO() [4/4]

```
gdcm::LO::LO (
    const Superclass & s,
    size\_type pos = 0,
    size\_type n = npos ) [inline]
```

### 10.174.4 Member Function Documentation

#### 10.174.4.1 IsValid()

```
bool gdcm::LO::IsValid ( ) const [inline]
```

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

- [gdcmLO.h](#)

## 10.175 gdcm::LookupTable Class Reference

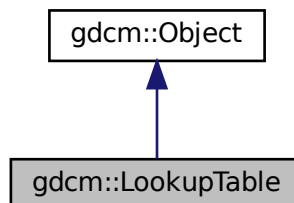
[LookupTable](#) class.

```
#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](#) ()
- void [Allocate](#) (unsigned short bitsample=8)  
*Allocate the LUT.*
- void [Clear](#) ()  
*Clear the LUT.*
- void [Decode](#) (std::istream &is, std::ostream &os) const  
*Decode the LUT.*
- bool [Decode](#) (char \*outputbuffer, size\_t outlen, const char \*inputbuffer, size\_t inlen) const
- 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:*
- void [Print](#) (std::ostream &) const
- 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.*

## Protected Attributes

- unsigned short [BitSample](#)
- bool [IncompleteLUT](#):1
- [LookupTableInternal](#) \* [Internal](#)

## Additional Inherited Members

### 10.175.1 Detailed Description

[LookupTable](#) class.

## 10.175.2 Member Enumeration Documentation

### 10.175.2.1 LookupTableType

enum `gdcm::LookupTable::LookupTableType`

#### Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

## 10.175.3 Constructor & Destructor Documentation

### 10.175.3.1 LookupTable() [1/2]

`gdcm::LookupTable::LookupTable ( )`

### 10.175.3.2 ~LookupTable()

`gdcm::LookupTable::~~LookupTable ( )`

### 10.175.3.3 LookupTable() [2/2]

`gdcm::LookupTable::LookupTable (`  
    `LookupTable const & lut ) [inline]`

## 10.175.4 Member Function Documentation

#### 10.175.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
    unsigned short bitsample = 8 )
```

Allocate the LUT.

#### 10.175.4.2 Clear()

```
void gdcm::LookupTable::Clear ( )
```

Clear the LUT.

#### 10.175.4.3 Decode() [1/2]

```
void gdcm::LookupTable::Decode (
    std::istream & is,
    std::ostream & os ) const
```

Decode the LUT.

#### 10.175.4.4 Decode() [2/2]

```
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

#### 10.175.4.5 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]
```

return the bit sample

#### 10.175.4.6 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
    unsigned char * rgba ) const
```

return the LUT as RGBA buffer

#### 10.175.4.7 GetLUT()

```
void gdcm::LookupTable::GetLUT (
    LookupTableType type,
    unsigned char * array,
    unsigned int & length ) const
```

Examples:

[ExtractImageRegionWithLUT.cs](#).

#### 10.175.4.8 GetLUTDescriptor()

```
void gdcm::LookupTable::GetLUTDescriptor (
    LookupTableType type,
    unsigned short & length,
    unsigned short & subscript,
    unsigned short & bitsize ) const
```

#### 10.175.4.9 GetLUTLength()

```
unsigned int gdcm::LookupTable::GetLUTLength (
    LookupTableType type ) const
```

#### 10.175.4.10 GetPointer()

```
const unsigned char* gdcm::LookupTable::GetPointer ( ) const
```

return a raw pointer to the LUT

#### 10.175.4.11 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

#### 10.175.4.12 Initialized()

```
bool gdcm::LookupTable::Initialized ( ) const
```

return whether the LUT has been initialized

#### 10.175.4.13 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

#### 10.175.4.14 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
    LookupTableType type,
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

Generic interface:

#### 10.175.4.15 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
    unsigned short length,
    unsigned short subscript,
    unsigned short bitsize )
```

RED / GREEN / BLUE specific:

#### 10.175.4.16 Print()

```
void gdcm::LookupTable::Print (
    std::ostream & ) const [inline], [virtual]
```

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

References [gdcm::terminal::blue](#), [gdcm::terminal::green](#), and [gdcm::terminal::red](#).

#### 10.175.4.17 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
    const unsigned char * blue,
    unsigned int length )
```

#### 10.175.4.18 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
    const unsigned char * green,
    unsigned int length )
```

#### 10.175.4.19 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
    LookupTableType type,
    const unsigned char * array,
    unsigned int length ) [virtual]
```

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

#### 10.175.4.20 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
    const unsigned char * red,
    unsigned int length )
```

#### 10.175.4.21 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
    const unsigned char * rgba )
```

Write the LUT as RGBA.

### 10.175.5 Member Data Documentation

#### 10.175.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample [protected]
```

#### 10.175.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT [protected]
```

#### 10.175.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal [protected]
```

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

- [gdcmLookupTable.h](#)

## 10.176 gdcm::Scanner::Itstr Struct Reference

```
#include <gdcmScanner.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.176.1 Member Function Documentation

#### 10.176.1.1 operator()

```
bool gdcm::Scanner::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

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

- [gdcmScanner.h](#)

### 10.177 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

#### Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 10.177.1 Member Function Documentation

#### 10.177.1.1 operator()

```
bool gdcm::StrictScanner::ltstr::operator() (
    const char * s1,
    const char * s2 ) const [inline]
```

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

- [gdcmStrictScanner.h](#)

### 10.178 gdcm::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](#) ()
- 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)

### 10.178.1 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](#)

### 10.178.2 Member Typedef Documentation

#### 10.178.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType
```

### 10.178.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcmm::Macro::MapModuleEntry
```

## 10.178.3 Constructor & Destructor Documentation

### 10.178.3.1 Macro()

```
gdcmm::Macro::Macro ( ) [inline]
```

References `gdcmm::operator<<()`.

## 10.178.4 Member Function Documentation

### 10.178.4.1 AddMacroEntry()

```
void gdcmm::Macro::AddMacroEntry (
    const Tag & tag,
    const MacroEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

### 10.178.4.2 Clear()

```
void gdcmm::Macro::Clear ( ) [inline]
```

### 10.178.4.3 FindMacroEntry()

```
bool gdcmm::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.

#### 10.178.4.4 GetMacroEntry()

```
const MacroEntry& gdcm::Macro::GetMacroEntry (
    const Tag & tag ) const
```

#### 10.178.4.5 GetName()

```
const char* gdcm::Macro::GetName ( ) const [inline]
```

#### 10.178.4.6 SetName()

```
void gdcm::Macro::SetName (
    const char * name ) [inline]
```

#### 10.178.4.7 Verify()

```
bool gdcm::Macro::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.178.5 Friends And Related Function Documentation

#### 10.178.5.1 operator<<

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

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

- [gdcmMacro.h](#)

## 10.179 gdcmmacros Class Reference

Class for representing a [Modules](#).

```
#include <gdcmmacros.h>
```

### Public Types

- typedef std::map< std::string, [Macro](#) > [ModuleMapType](#)

### Public Member Functions

- [Macros](#) ()
- 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)

### 10.179.1 Detailed Description

Class for representing a [Modules](#).

#### Note

bla

#### See also

[Module](#)

#### Examples:

[TraverseModules.cxx](#).

### 10.179.2 Member Typedef Documentation

### 10.179.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcmmacros::ModuleMapType
```

## 10.179.3 Constructor & Destructor Documentation

### 10.179.3.1 Macros()

```
gdcmmacros::Macros::Macros ( ) [inline]
```

References `gdcmmacros::operator<<()`.

## 10.179.4 Member Function Documentation

### 10.179.4.1 AddMacro()

```
void gdcmmacros::AddMacro (
    const char * ref,
    const Macro & module ) [inline]
```

### 10.179.4.2 Clear()

```
void gdcmmacros::Clear ( ) [inline]
```

### 10.179.4.3 GetMacro()

```
const Macro& gdcmmacros::GetMacro (
    const char * name ) const [inline]
```

### 10.179.4.4 IsEmpty()

```
bool gdcmmacros::IsEmpty ( ) const [inline]
```

## 10.179.5 Friends And Related Function Documentation

### 10.179.5.1 operator<<

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

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

- [gdcmMacros.h](#)

## 10.180 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

### 10.180.1 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)

### 10.180.2 Constructor & Destructor Documentation

### 10.180.2.1 MaximumLengthSub()

```
gdcmm::network::MaximumLengthSub::MaximumLengthSub ( )
```

## 10.180.3 Member Function Documentation

### 10.180.3.1 GetMaximumLength()

```
uint32_t gdcmm::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]
```

References Print(), and SetMaximumLength().

### 10.180.3.2 Print()

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

Referenced by GetMaximumLength().

### 10.180.3.3 Read()

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

### 10.180.3.4 SetMaximumLength()

```
void gdcmm::network::MaximumLengthSub::SetMaximumLength (
    uint32_t maximumlength )
```

Referenced by GetMaximumLength().

### 10.180.3.5 Size()

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

### 10.180.3.6 Write()

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

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

- [gdcmmMaximumLengthSub.h](#)

## 10.181 gdcmm::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcmmMD5.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*

### 10.181.1 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



## 10.181.2 Member Function Documentation

### 10.181.2.1 Compute()

```
static bool gdcm::MD5::Compute (
    const char * buffer,
    size_t buf_len,
    char digest_str[33] ) [static]
```

### 10.181.2.2 ComputeFile()

```
static 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](#)

## 10.182 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,
    LegacyConvertedEnhancedMRIImageStorage,
    LegacyConvertedEnhancedPETImageStorage,
    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)

### 10.182.1 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\\_Image\\_Writer.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.↵  
cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [TestReader.cxx](#).

### 10.182.2 Member Enumeration Documentation

#### 10.182.2.1 MSType

```
enum gdcm::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	
XRayRadiofluoroscopingImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	

## Enumerator

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	

## Enumerator

XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRIImageStorage	
LegacyConvertedEnhancedPETImageStorage	
MS_END	

## Examples:

[GenerateStandardSOPClasses.cxx](#).

## 10.182.2.2 ObjectType

```
enum gdcm::MediaStorage::ObjectType
```

## Enumerator

NoObject	
Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

## 10.182.3 Constructor &amp; Destructor Documentation

## 10.182.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
    MType type = MS_END ) [inline]
```

## 10.182.4 Member Function Documentation

#### 10.182.4.1 GetModality()

```
const char* gdcm::MediaStorage::GetModality ( ) const
```

#### 10.182.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const
```

#### 10.182.4.3 GetMSString()

```
static const char* gdcm::MediaStorage::GetMSString (
    MType ts ) [static]
```

Return the Media [String](#) associated. Will return NULL for MS\_END.

##### Examples:

[GenerateStandardSOPClasses.cxx](#).

Referenced by `gdcm::operator<<()`.

#### 10.182.4.4 GetMSType()

```
static MType gdcm::MediaStorage::GetMSType (
    const char * str ) [static]
```

##### Examples:

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

#### 10.182.4.5 GetNumberOfModality()

```
static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]
```



#### 10.182.4.6 GetNumberOfMSString()

```
static unsigned int gdcM::MediaStorage::GetNumberOfMSString ( ) [static]
```

#### 10.182.4.7 GetNumberOfMSType()

```
static unsigned int gdcM::MediaStorage::GetNumberOfMSType ( ) [static]
```

#### 10.182.4.8 GetString()

```
const char* gdcM::MediaStorage::GetString ( ) const
```

Return the Media [String](#) of the object.

Examples:

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

#### 10.182.4.9 GuessFromModality()

```
void gdcM::MediaStorage::GuessFromModality (
    const char * modality,
    unsigned int dimension = 2 )
```

#### 10.182.4.10 IsImage()

```
static 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](#).

#### 10.182.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined ( ) const [inline]
```

Examples:

[TestReader.cxx](#).

#### 10.182.4.12 operator MType()

```
gdcm::MediaStorage::operator MType ( ) const [inline]
```

References `gdcm::operator<<()`.

#### 10.182.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
    DataSet const & ds )
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

#### 10.182.4.14 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:

[gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), and [TestReader.cxx](#).

#### 10.182.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
    FileMetaInformation const & fmi )
```

#### 10.182.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
    DataSet const & ds )
```

#### 10.182.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
    DataSet const & ds ) [protected]
```

### 10.182.5 Friends And Related Function Documentation

#### 10.182.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const MediaStorage & ms ) [friend]
```

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

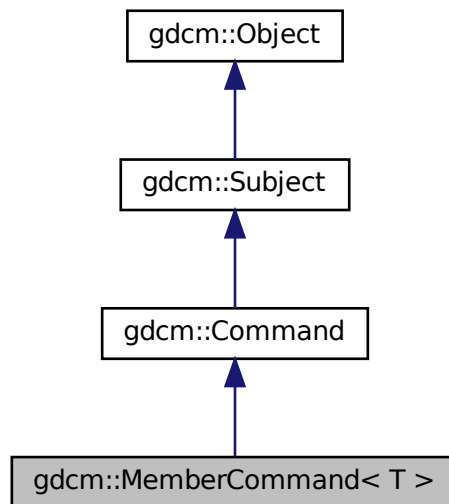
- [gdcmMediaStorage.h](#)

## 10.183 gdcm::MemberCommand< T > Class Template Reference

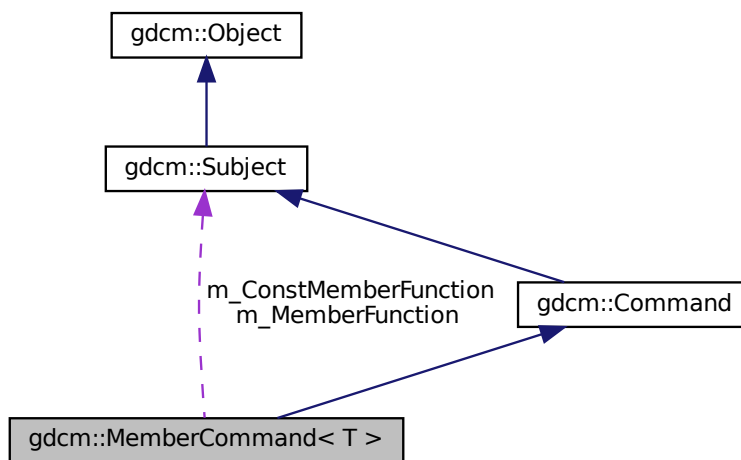
[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for `gdc::MemberCommand< T >`:



Collaboration diagram for `gdc::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

- virtual void Execute (Subject \*caller, const Event &event)
- virtual void Execute (const Subject \*caller, const Event &event)
- void SetCallbackFunction (T \*object, TMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T \*object, TConstMemberFunctionPointer memberFunction)

### Static Public Member Functions

- static SmartPointer< MemberCommand > New ()

### Protected Member Functions

- MemberCommand ()
- virtual ~MemberCommand ()

### Protected Attributes

- TConstMemberFunctionPointer m\_ConstMemberFunction
- TMemberFunctionPointer m\_MemberFunction
- T \* m\_This

## 10.183.1 Detailed Description

```
template<class T>
class gdcmm::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.

## 10.183.2 Member Typedef Documentation

### 10.183.2.1 Self

```
template<class T >
typedef MemberCommand gdcmm::MemberCommand< T >::Self
```

Standard class typedefs.

### 10.183.2.2 TConstMemberFunctionPointer

```
template<class T >
typedef void(T::* gdcM::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const Event &)
```

### 10.183.2.3 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

## 10.183.3 Constructor & Destructor Documentation

### 10.183.3.1 MemberCommand()

```
template<class T >
gdcM::MemberCommand< T >::MemberCommand ( ) [inline], [protected]
```

### 10.183.3.2 ~MemberCommand()

```
template<class T >
virtual gdcM::MemberCommand< T >::~~MemberCommand ( ) [inline], [protected], [virtual]
```

## 10.183.4 Member Function Documentation

### 10.183.4.1 Execute() [1/2]

```
template<class T >
virtual void gdcM::MemberCommand< T >::Execute (
    Subject * caller,
    const Event & event ) [inline], [virtual]
```

Invoke the member function.

Implements [gdcM::Command](#).

**10.183.4.2 Execute()** [2/2]

```
template<class T >
virtual void gdcM::MemberCommand< T >::Execute (
    const Subject * caller,
    const Event & event ) [inline], [virtual]
```

Invoke the member function with a const object.

Implements [gdcM::Command](#).

**10.183.4.3 New()**

```
template<class T >
static SmartPointer<MemberCommand> gdcM::MemberCommand< T >::New ( ) [inline], [static]
```

Method for creation through the object factory.

**10.183.4.4 SetCallbackFunction()** [1/2]

```
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.

**10.183.4.5 SetCallbackFunction()** [2/2]

```
template<class T >
void gdcM::MemberCommand< T >::SetCallbackFunction (
    T * object,
    TConstMemberFunctionPointer memberFunction ) [inline]
```

**10.183.5 Member Data Documentation****10.183.5.1 m\_ConstMemberFunction**

```
template<class T >
TConstMemberFunctionPointer gdcM::MemberCommand< T >::m_ConstMemberFunction [protected]
```

### 10.183.5.2 m\_MemberFunction

```
template<class T >  
TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]
```

### 10.183.5.3 m\_This

```
template<class T >  
T* gdcm::MemberCommand< T >::m_This [protected]
```

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

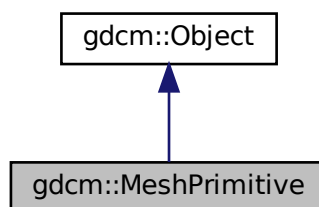
- [gdcmCommand.h](#)

## 10.184 gdcm::MeshPrimitive Class Reference

This class defines surface mesh primitives.

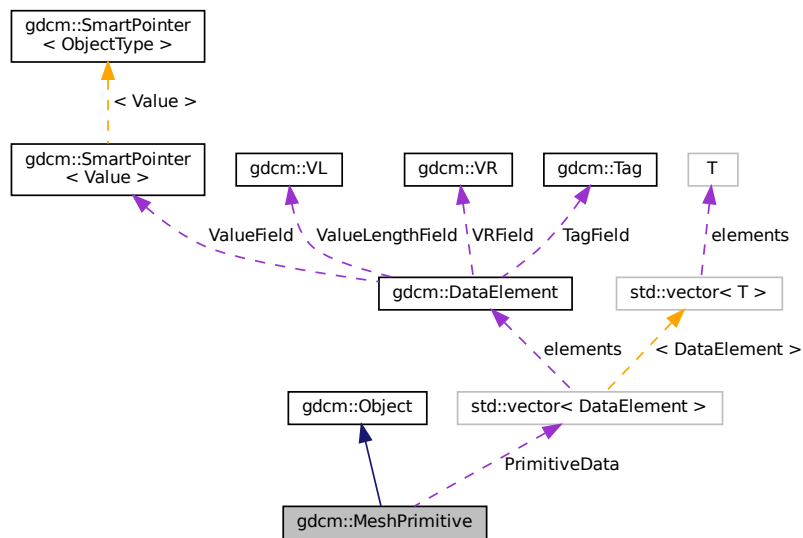
```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for gdcm::MeshPrimitive:





Collaboration diagram for gdcm::MeshPrimitive:



## 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` ()
- virtual `~MeshPrimitive` ()
- void `AddPrimitiveData` (`DataElement` const &de)
- unsigned int `GetNumberOfPrimitivesData` () const
- 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` ()

- [MPTYPE GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPTYPE](#) type)

### 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

#### 10.184.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

#### 10.184.2 Member Typedef Documentation

##### 10.184.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

#### 10.184.3 Member Enumeration Documentation

##### 10.184.3.1 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	
MPTYPE_END	

## 10.184.4 Constructor & Destructor Documentation

### 10.184.4.1 MeshPrimitive()

```
gdcm::MeshPrimitive::MeshPrimitive ( )
```

### 10.184.4.2 ~MeshPrimitive()

```
virtual gdcm::MeshPrimitive::~~MeshPrimitive ( ) [virtual]
```

## 10.184.5 Member Function Documentation

### 10.184.5.1 AddPrimitiveData()

```
void gdcm::MeshPrimitive::AddPrimitiveData (
    DataElement const & de )
```

### 10.184.5.2 GetMPTYPE()

```
static MPTYPE gdcm::MeshPrimitive::GetMPTYPE (
    const char * type ) [static]
```

**10.184.5.3 GetMPTypeString()**

```
static const char* gdcm::MeshPrimitive::GetMPTypeString (
    const MPType type ) [static]
```

**10.184.5.4 GetNumberOfPrimitivesData()**

```
unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const
```

**10.184.5.5 GetPrimitiveData()** [1/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const
```

**10.184.5.6 GetPrimitiveData()** [2/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )
```

**10.184.5.7 GetPrimitiveData()** [3/4]

```
const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx ) const
```

**10.184.5.8 GetPrimitiveData()** [4/4]

```
DataElement& gdcm::MeshPrimitive::GetPrimitiveData (
    const unsigned int idx )
```

**10.184.5.9 GetPrimitivesData()** [1/2]

```
const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const
```

#### 10.184.5.10 GetPrimitivesData() [2/2]

```
PrimitivesData& gdcmmeshprimitive::GetPrimitivesData ( )
```

#### 10.184.5.11 GetPrimitiveType()

```
MPTyp gdcmmeshprimitive::GetPrimitiveType ( ) const
```

#### 10.184.5.12 SetPrimitiveData() [1/2]

```
void gdcmmeshprimitive::SetPrimitiveData (
    DataElement const & de )
```

#### 10.184.5.13 SetPrimitiveData() [2/2]

```
void gdcmmeshprimitive::SetPrimitiveData (
    const unsigned int idx,
    DataElement const & de )
```

#### 10.184.5.14 SetPrimitivesData()

```
void gdcmmeshprimitive::SetPrimitivesData (
    PrimitivesData const & DEs )
```

#### 10.184.5.15 SetPrimitiveType()

```
void gdcmmeshprimitive::SetPrimitiveType (
    const MPTyp type )
```

### 10.184.6 Member Data Documentation

### 10.184.6.1 PrimitiveData

```
PrimitivesData gdcM::MeshPrimitive::PrimitiveData [protected]
```

### 10.184.6.2 PrimitiveType

```
MPTType gdcM::MeshPrimitive::PrimitiveType [protected]
```

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

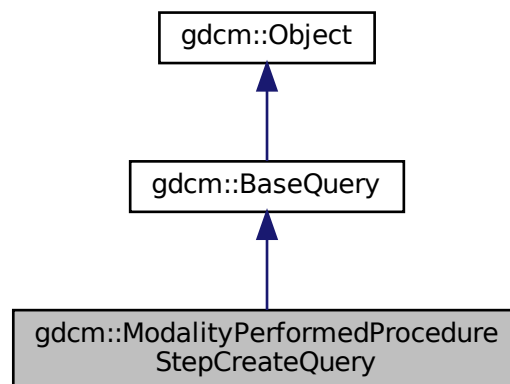
- [gdcMMeshPrimitive.h](#)

## 10.185 gdcM::ModalityPerformedProcedureStepCreateQuery Class Reference

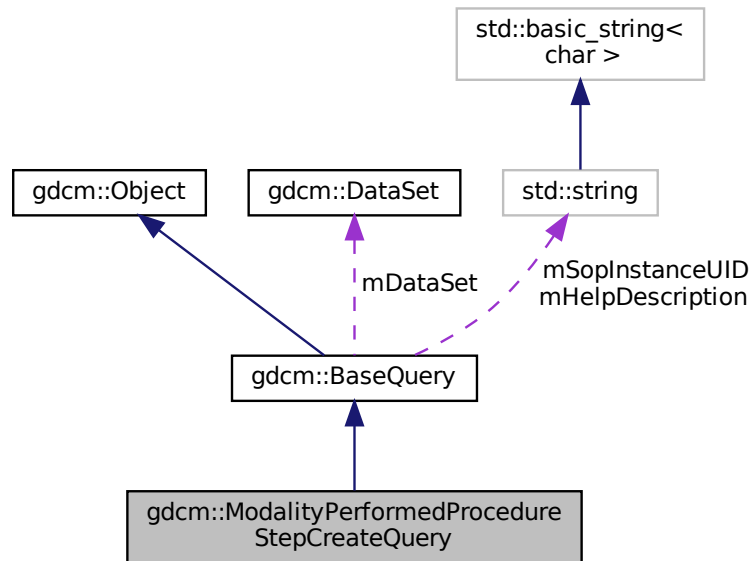
[ModalityPerformedProcedureStepCreateQuery.](#)

```
#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
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.185.1 Detailed Description

[ModalityPerformedProcedureStepCreateQuery](#).

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

## 10.185.2 Constructor & Destructor Documentation

### 10.185.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
    const std::string & iSopInstanceUID )
```

## 10.185.3 Member Function Documentation

### 10.185.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.185.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet ( ) const
```

### 10.185.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
    bool inStrict = true ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

## 10.185.4 Friends And Related Function Documentation



## 10.185.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

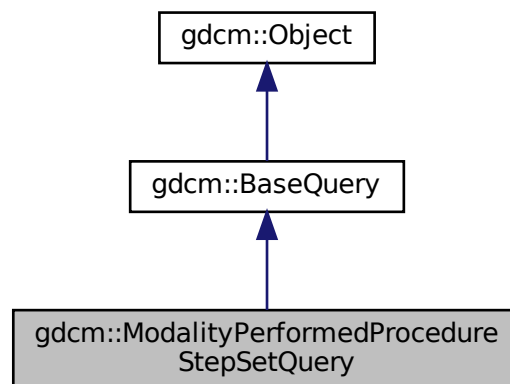
- [gdcmModalityPerformedProcedureStepCreateQuery.h](#)

## 10.186 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

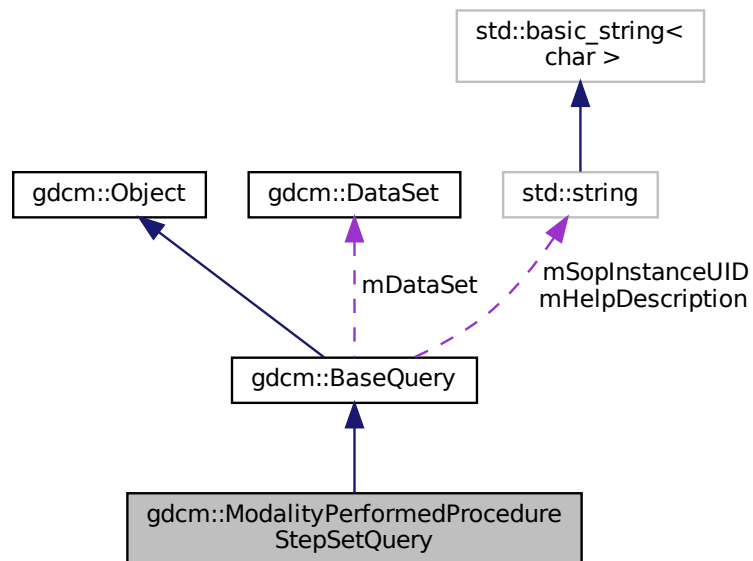
[ModalityPerformedProcedureStepSetQuery](#).

```
#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
- [gdcm::DataSet GetRequiredDataSet](#) () const
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.186.1 Detailed Description

[ModalityPerformedProcedureStepSetQuery](#).

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

## 10.186.2 Constructor & Destructor Documentation

### 10.186.2.1 ModalityPerformedProcedureStepSetQuery()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
    const std::string & iSopInstanceUID )
```

## 10.186.3 Member Function Documentation

### 10.186.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.186.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet ( ) const
```

### 10.186.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
    bool inStrict = true ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

## 10.186.4 Friends And Related Function Documentation

#### 10.186.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

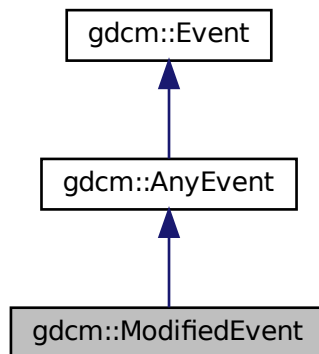
The documentation for this class was generated from the following file:

- [gdcmModalityPerformedProcedureStepSetQuery.h](#)

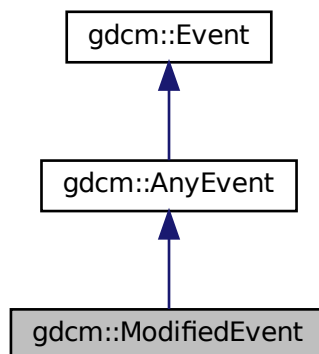
### 10.187 gdcm::ModifiedEvent Class Reference

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

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



## Additional Inherited Members

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

- [gdcmmodule.h](#)

## 10.188 gdcmmodule Class Reference

Class for representing a [Module](#).

```
#include <gdcmmodule.h>
```

### Public Types

- typedef std::vector< std::string > [ArrayIncludeMacrosType](#)
- typedef std::map< [Tag](#), [ModuleEntry](#) > [MapModuleEntry](#)

### Public Member Functions

- [Module](#) ()
- 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 &macros, const [Tag](#) &tag) const
- const [ModuleEntry](#) & [GetModuleEntryInMacros](#) ([Macros](#) const &macros, 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)

### 10.188.1 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](#).

## 10.188.2 Member Typedef Documentation

### 10.188.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcM::Module::ArrayIncludeMacrosType
```

### 10.188.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcM::Module::MapModuleEntry
```

## 10.188.3 Constructor & Destructor Documentation

### 10.188.3.1 Module()

```
gdcM::Module::Module ( ) [inline]
```

References [gdcM::operator<<\(\)](#).

## 10.188.4 Member Function Documentation

### 10.188.4.1 AddMacro()

```
void gdcM::Module::AddMacro (  
    const char * include ) [inline]
```

### 10.188.4.2 AddModuleEntry()

```
void gdcM::Module::AddModuleEntry (  
    const Tag & tag,  
    const ModuleEntry & module ) [inline]
```

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

#### 10.188.4.3 Clear()

```
void gdcmm::Module::Clear ( ) [inline]
```

#### 10.188.4.4 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](#).

#### 10.188.4.5 GetModuleEntryInMacros()

```
const ModuleEntry& gdcmm::Module::GetModuleEntryInMacros (
    Macros const & macros,
    const Tag & tag ) const
```

Examples:

[TraverseModules.cxx](#).

#### 10.188.4.6 GetName()

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

#### 10.188.4.7 SetName()

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

#### 10.188.4.8 Verify()

```
bool gdcm::Module::Verify (
    const DataSet & ds,
    Usage const & usage ) const
```

### 10.188.5 Friends And Related Function Documentation

#### 10.188.5.1 operator<<

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

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

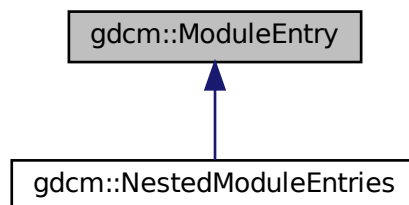
- [gdcmModule.h](#)

## 10.189 gdcm::ModuleEntry Class Reference

Class for representing a [ModuleEntry](#).

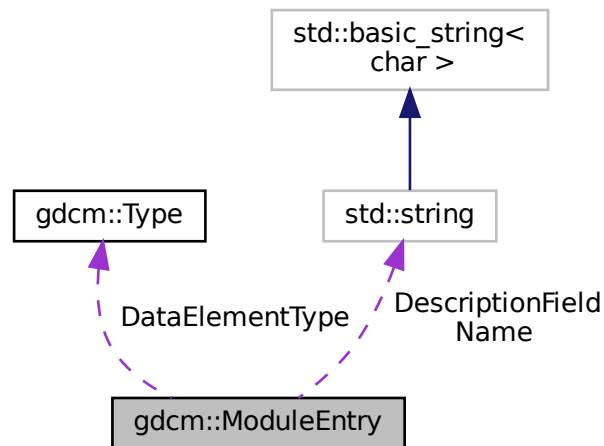
```
#include <gdcmModuleEntry.h>
```

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](#) ()
- 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)

### 10.189.1 Detailed Description

Class for representing a [ModuleEntry](#).

#### Note

bla

#### See also

[DictEntry](#)

#### Examples:

[TraverseModules.cxx](#).

### 10.189.2 Member Typedef Documentation

#### 10.189.2.1 Description

```
typedef std::string gdcmm::ModuleEntry::Description
```

### 10.189.3 Constructor & Destructor Documentation

#### 10.189.3.1 ModuleEntry()

```
gdcmm::ModuleEntry::ModuleEntry (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

References [gdcmm::Type::GetTypeType\(\)](#).

#### 10.189.3.2 ~ModuleEntry()

```
virtual gdcmm::ModuleEntry::~~ModuleEntry ( ) [inline], [virtual]
```

References [gdcmm::operator<<\(\)](#).

## 10.189.4 Member Function Documentation

### 10.189.4.1 GetDescription()

```
const Description& gdcmmoduleentry::GetDescription ( ) const [inline]
```

### 10.189.4.2 GetName()

```
const char* gdcmmoduleentry::GetName ( ) const [inline]
```

### 10.189.4.3 GetType()

```
const Type& gdcmmoduleentry::GetType ( ) const [inline]
```

#### Examples:

[TraverseModules.cxx](#).

### 10.189.4.4 SetDescription()

```
void gdcmmoduleentry::SetDescription (
    const char * d ) [inline]
```

### 10.189.4.5 SetName()

```
void gdcmmoduleentry::SetName (
    const char * name ) [inline]
```

### 10.189.4.6 SetType()

```
void gdcmmoduleentry::SetType (
    const Type & type ) [inline]
```

## 10.189.5 Friends And Related Function Documentation

### 10.189.5.1 operator<<

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

## 10.189.6 Member Data Documentation

### 10.189.6.1 DataElementType

`Type` `gdcM::ModuleEntry::DataElementType` [protected]

Referenced by `gdcM::operator<<()`.

### 10.189.6.2 DescriptionField

`Description` `gdcM::ModuleEntry::DescriptionField` [protected]

Referenced by `gdcM::operator<<()`.

### 10.189.6.3 Name

`std::string` `gdcM::ModuleEntry::Name` [protected]

Referenced by `gdcM::operator<<()`.

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

- [gdcMModuleEntry.h](#)

## 10.190 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](#) ()
- 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)

### 10.190.1 Detailed Description

Class for representing a [Modules](#).

#### Note

bla

#### See also

[Module](#)

#### Examples:

[TraverseModules.cxx](#).

### 10.190.2 Member Typedef Documentation

### 10.190.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcM::Modules::ModuleMapType
```

## 10.190.3 Constructor & Destructor Documentation

### 10.190.3.1 Modules()

```
gdcM::Modules::Modules ( ) [inline]
```

References `gdcM::operator<<()`.

## 10.190.4 Member Function Documentation

### 10.190.4.1 AddModule()

```
void gdcM::Modules::AddModule (
    const char * ref,
    const Module & module ) [inline]
```

### 10.190.4.2 Clear()

```
void gdcM::Modules::Clear ( ) [inline]
```

### 10.190.4.3 GetModule()

```
const Module& gdcM::Modules::GetModule (
    const char * name ) const [inline]
```

#### Examples:

[TraverseModules.cxx](#).

#### 10.190.4.4 isEmpty()

```
bool gdcm::Modules::IsEmpty ( ) const [inline]
```

### 10.190.5 Friends And Related Function Documentation

#### 10.190.5.1 operator<<

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

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

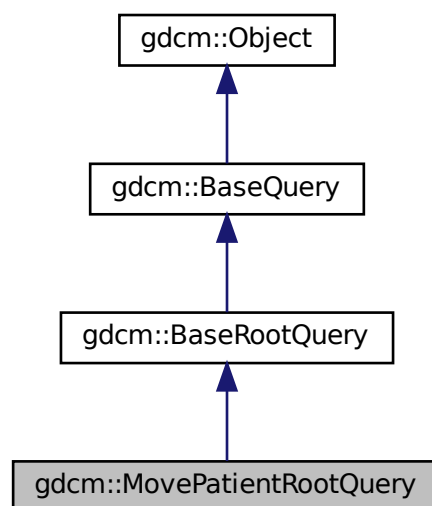
- [gdcmModules.h](#)

## 10.191 gdcm::MovePatientRootQuery Class Reference

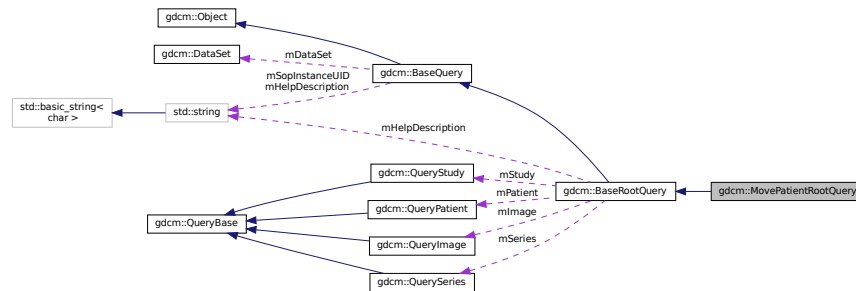
[MovePatientRootQuery](#).

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for `gdcm::MovePatientRootQuery`:



## Public Member Functions

- [MovePatientRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- `std::vector< Tag > GetTagListByLevel` (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.191.1 Detailed Description

[MovePatientRootQuery](#).

contains: the class which will produce a dataset for c-move with patient root

### 10.191.2 Constructor & Destructor Documentation

#### 10.191.2.1 MovePatientRootQuery()

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ( )
```



### 10.191.3 Member Function Documentation

#### 10.191.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

#### 10.191.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::MovePatientRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [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](#).

#### 10.191.3.3 InitializeDataSet()

```
void gdcm::MovePatientRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [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 dcmrk

Implements [gdcm::BaseRootQuery](#).

#### 10.191.3.4 ValidateQuery()

```
bool gdcm::MovePatientRootQuery::ValidateQuery (
    bool inStrict = true ) const [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](#).

#### 10.191.4 Friends And Related Function Documentation

##### 10.191.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

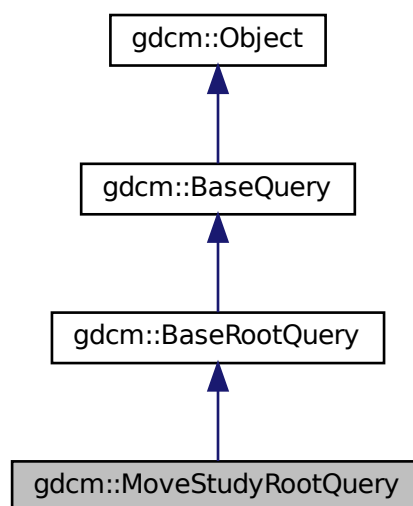
- [gdcmMovePatientRootQuery.h](#)

#### 10.192 gdcm::MoveStudyRootQuery Class Reference

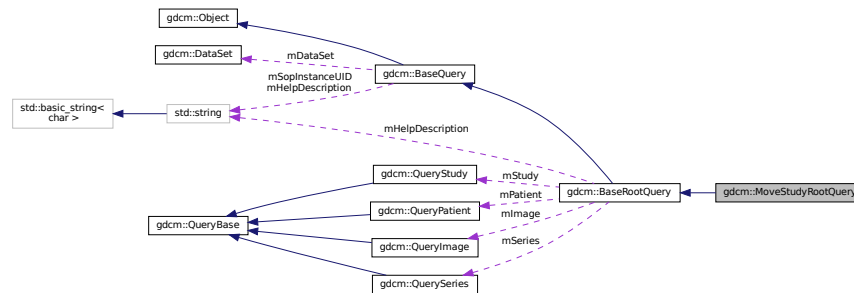
[MoveStudyRootQuery](#).

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



## Public Member Functions

- [MoveStudyRootQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.192.1 Detailed Description

[MoveStudyRootQuery](#).

contains: the class which will produce a dataset for C-MOVE with study root

### 10.192.2 Constructor & Destructor Documentation

#### 10.192.2.1 MoveStudyRootQuery()

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )
```

### 10.192.3 Member Function Documentation

#### 10.192.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcM::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcM::BaseQuery](#).

#### 10.192.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcM::MoveStudyRootQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [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](#).

#### 10.192.3.3 InitializeDataSet()

```
void gdcM::MoveStudyRootQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [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 dcm4k

Implements [gdcM::BaseRootQuery](#).

#### 10.192.3.4 ValidateQuery()

```
bool gdcM::MoveStudyRootQuery::ValidateQuery (
    bool inStrict = true ) const [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](#).

## 10.192.4 Friends And Related Function Documentation

### 10.192.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

- [gdcmMoveStudyRootQuery.h](#)

## 10.193 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)

### 10.193.1 Detailed Description

Class for [MrProtocol](#).

Examples:

[MrProtocol.cxx](#).

### 10.193.2 Constructor & Destructor Documentation

#### 10.193.2.1 MrProtocol()

```
gdcm::MrProtocol::MrProtocol ( )
```

#### 10.193.2.2 ~MrProtocol()

```
gdcm::MrProtocol::~~MrProtocol ( )
```

### 10.193.3 Member Function Documentation

#### 10.193.3.1 FindMrProtocolByName()

```
bool gdcm::MrProtocol::FindMrProtocolByName (
    const char * name ) const
```

#### 10.193.3.2 GetMrProtocolByName()

```
const char* gdcm::MrProtocol::GetMrProtocolByName (
    const char * name ) const
```

### 10.193.3.3 GetSliceArray()

```
bool gdcm::MrProtocol::GetSliceArray (
    MrProtocol::SliceArray & sa ) const
```

### 10.193.3.4 GetVersion()

```
int gdcm::MrProtocol::GetVersion ( ) const
```

### 10.193.3.5 Load()

```
bool gdcm::MrProtocol::Load (
    const ByteValue * bv,
    const char * str,
    int version )
```

### 10.193.3.6 Print()

```
void gdcm::MrProtocol::Print (
    std::ostream & os ) const
```

Referenced by `gdcm::operator<<()`.

## 10.193.4 Friends And Related Function Documentation

### 10.193.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const MrProtocol & d ) [friend]
```

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

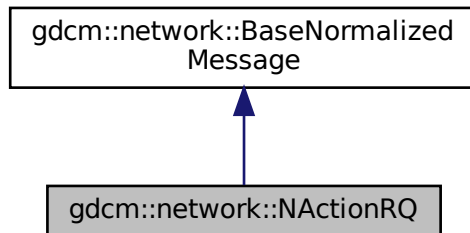
- [gdcmMrProtocol.h](#)

## 10.194 gdcmm::network::NActionRQ Class Reference

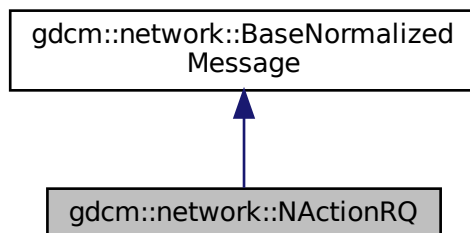
[NActionRQ](#).

```
#include <gdcmmNActionMessages.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)

### 10.194.1 Detailed Description

[NActionRQ](#).

this file defines the messages for the NAction action



## 10.194.2 Member Function Documentation

### 10.194.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NActionRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

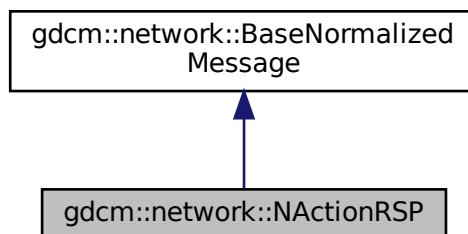
- [gdcmNActionMessages.h](#)

## 10.195 gdcm::network::NActionRSP Class Reference

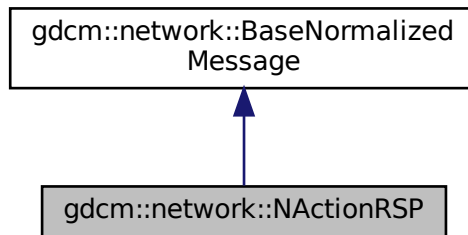
[NActionRSP](#) this file defines the messages for the NAction action.

```
#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)`

### 10.195.1 Detailed Description

[NActionRSP](#) this file defines the messages for the NAction action.

### 10.195.2 Member Function Documentation

#### 10.195.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NActionRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

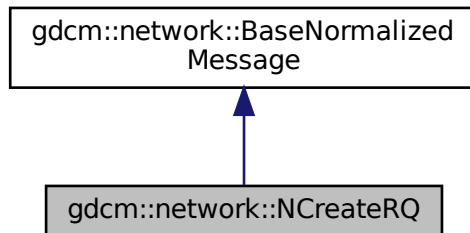
- [gdcmNActionMessages.h](#)

## 10.196 gdcm::network::NCreateRQ Class Reference

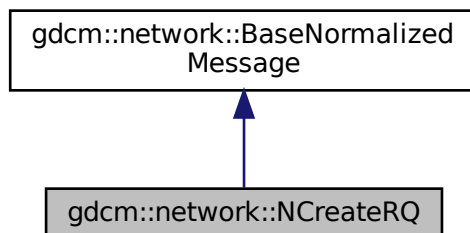
[NCreateRQ](#).

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRQ:



Collaboration diagram for gdcm::network::NCreateRQ:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseQuery](#) \*inQuery)

### 10.196.1 Detailed Description

[NCreateRQ](#).

this file defines the messages for the ncreate action

## 10.196.2 Member Function Documentation

### 10.196.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcmm::network::NCreateRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

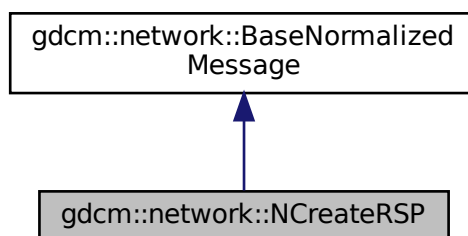
- [gdcmmNCreateMessages.h](#)

## 10.197 gdcmm::network::NCreateRSP Class Reference

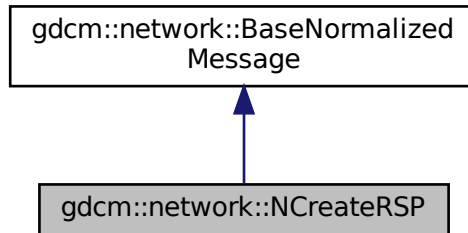
[NCreateRSP](#) this file defines the messages for the ncreate action.

```
#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)`

### 10.197.1 Detailed Description

[NCreateRSP](#) this file defines the messages for the ncreate action.

### 10.197.2 Member Function Documentation

#### 10.197.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NCreateRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

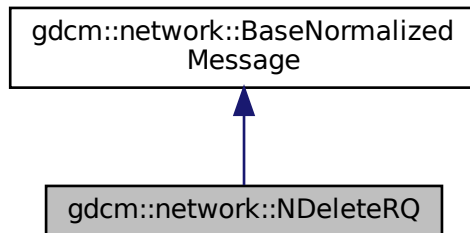
- [gdcmNCreateMessages.h](#)

## 10.198 gdcmm::network::NDeleteRQ Class Reference

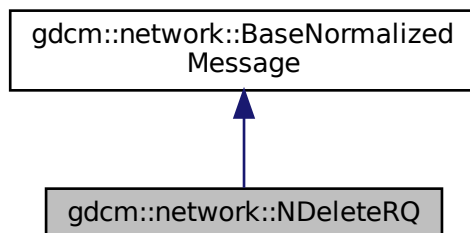
[NDeleteRQ](#).

```
#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)

### 10.198.1 Detailed Description

[NDeleteRQ](#).

this file defines the messages for the ndelete action

## 10.198.2 Member Function Documentation

### 10.198.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NDeleteRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

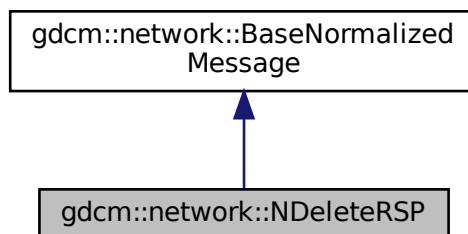
- [gdcmNDeleteMessages.h](#)

## 10.199 gdcm::network::NDeleteRSP Class Reference

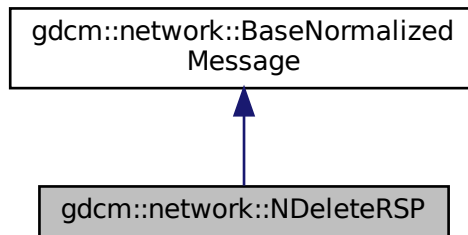
[NDeleteRSP](#) this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for `gdcm::network::NDeleteRSP`:



Collaboration diagram for `gdcmm::network::NDeleteRSP`:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### 10.199.1 Detailed Description

[NDeleteRSP](#) this file defines the messages for the ndelete action.

### 10.199.2 Member Function Documentation

#### 10.199.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcmm::network::NDeleteRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

- [gdcmmNDeleteMessages.h](#)

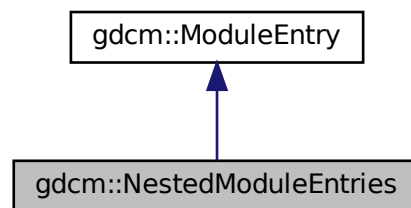


## 10.200 gdcm::NestedModuleEntries Class Reference

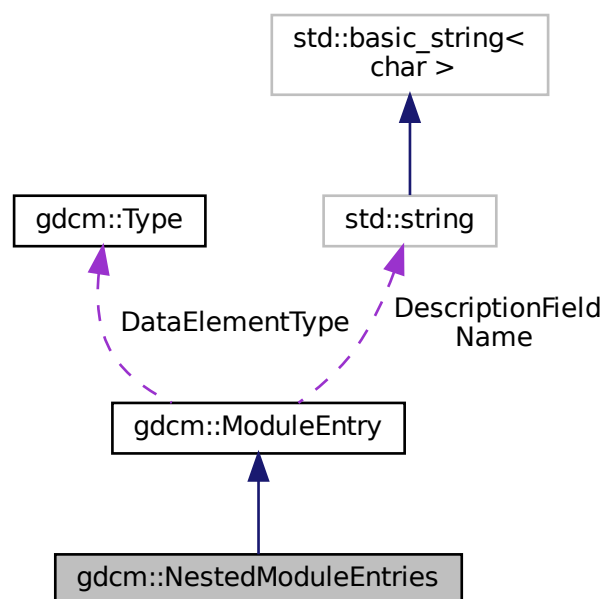
Class for representing a [NestedModuleEntries](#).

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



## Public Types

- `typedef std::vector< ModuleEntry >::size_type SizeType`

## Public Member Functions

- `NestedModuleEntries (const char *name="", const char *type="3", const char *description="")`
- `void AddModuleEntry (const ModuleEntry &me)`
- `const ModuleEntry & GetModuleEntry (SizeType idx) const`
- `ModuleEntry & GetModuleEntry (SizeType idx)`
- `SizeType GetNumberOfModuleEntries ()`

## Friends

- `std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

## Additional Inherited Members

### 10.200.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

#### Note

bla

#### See also

[ModuleEntry](#)

### 10.200.2 Member Typedef Documentation

#### 10.200.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcmm::NestedModuleEntries::SizeType
```

### 10.200.3 Constructor & Destructor Documentation

### 10.200.3.1 NestedModuleEntries()

```
gdcm::NestedModuleEntries::NestedModuleEntries (
    const char * name = "",
    const char * type = "3",
    const char * description = "" ) [inline]
```

References `gdcm::operator<<()`.

## 10.200.4 Member Function Documentation

### 10.200.4.1 AddModuleEntry()

```
void gdcm::NestedModuleEntries::AddModuleEntry (
    const ModuleEntry & me ) [inline]
```

### 10.200.4.2 GetModuleEntry() [1/2]

```
const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) const [inline]
```

### 10.200.4.3 GetModuleEntry() [2/2]

```
ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (
    SizeType idx ) [inline]
```

### 10.200.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]
```

## 10.200.5 Friends And Related Function Documentation

### 10.200.5.1 operator<<

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

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

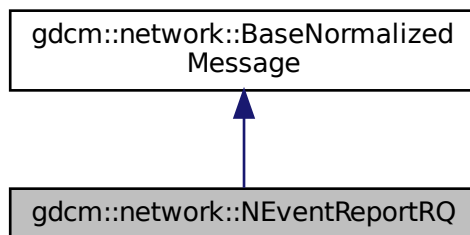
- [gdcmNestedModuleEntries.h](#)

## 10.201 gdcm::network::NEventReportRQ Class Reference

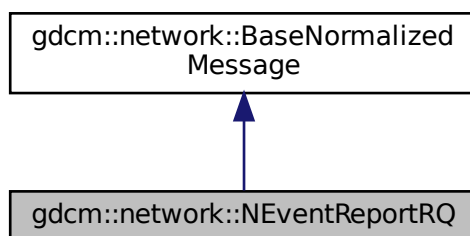
[NEventReportRQ](#).

```
#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`)

### 10.201.1 Detailed Description

[NEventReportRQ](#).

this file defines the messages for the neventreport action

### 10.201.2 Member Function Documentation

#### 10.201.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NEventReportRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

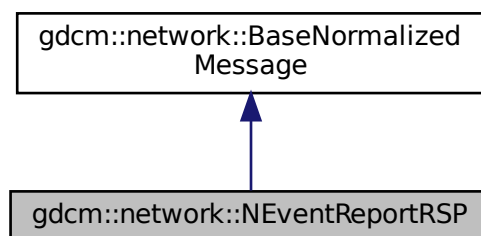
- [gdcmNEventReportMessages.h](#)

## 10.202 gdcm::network::NEventReportRSP Class Reference

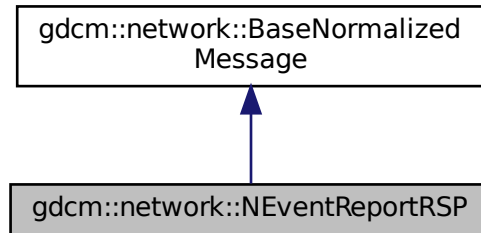
[NEventReportRSP](#) this file defines the messages for the neventreport action.

```
#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)`

### 10.202.1 Detailed Description

[NEventReportRSP](#) this file defines the messages for the neventreport action.

### 10.202.2 Member Function Documentation

#### 10.202.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NEventReportRSP::ConstructPDVByDataSet (  
    const DataSet * inDataSet )
```

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

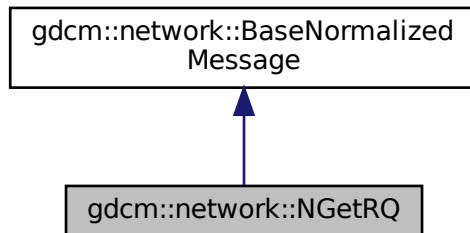
- [gdcmNEventReportMessages.h](#)

## 10.203 gdcm::network::NGetRQ Class Reference

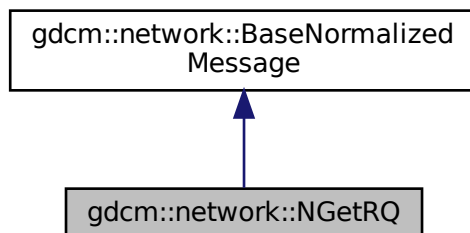
[NGetRQ](#).

```
#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)

### 10.203.1 Detailed Description

[NGetRQ](#).

this file defines the messages for the nget action

## 10.203.2 Member Function Documentation

### 10.203.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NGetRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

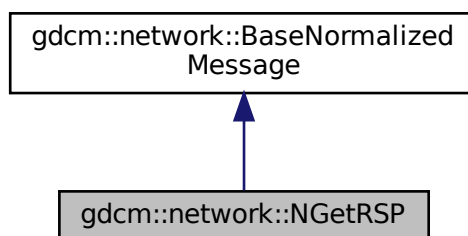
- [gdcmNGetMessages.h](#)

## 10.204 gdcm::network::NGetRSP Class Reference

[NGetRSP](#) this file defines the messages for the nget action.

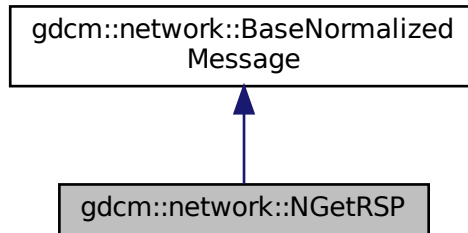
```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for `gdcm::network::NGetRSP`:





Collaboration diagram for gdcm::network::NGetRSP:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### 10.204.1 Detailed Description

[NGetRSP](#) this file defines the messages for the nget action.

### 10.204.2 Member Function Documentation

#### 10.204.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcm::network::NGetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

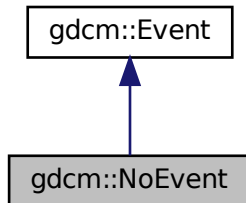
The documentation for this class was generated from the following file:

- [gdcmNGetMessages.h](#)

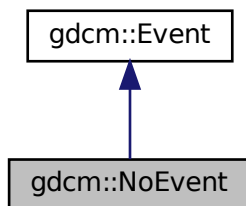
## 10.205 gdcm::NoEvent Class Reference

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

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



### Additional Inherited Members

#### 10.205.1 Detailed Description

Define some common GDCM events

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

- [gdcmEvent.h](#)

## 10.206 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)

### 10.206.1 Member Function Documentation

#### 10.206.1.1 ConstructNAction()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNAction (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.206.1.2 ConstructNCreate()

```
static std::vector<PresentationDataValue> gdcm::network::NormalizedMessageFactory::ConstructNCreate (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.206.1.3 ConstructNDelete()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNDelete (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.206.1.4 ConstructNEventReport()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNEventReport (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.206.1.5 ConstructNGet()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNGet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

### 10.206.1.6 ConstructNSet()

```
static std::vector<PresentationDataValue> gdcM::network::NormalizedMessageFactory::ConstructNSet (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

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

- [gdcMNormalizedMessageFactory.h](#)

## 10.207 gdcM::NormalizedNetworkFunctions Class Reference

Normalized Network Functions.

```
#include <gdcMNormalizedNetworkFunctions.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)

### 10.207.1 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

### 10.207.2 Member Function Documentation

#### 10.207.2.1 ConstructQuery()

```
static BaseQuery* gdcmm::NormalizedNetworkFunctions::ConstructQuery (
    const std::string & sopInstanceUID,
    const DataSet & queryds,
    ENQueryType queryType = eCreateMMPS ) [static]
```

### 10.207.2.2 NAction()

```
static bool gdcm::NormalizedNetworkFunctions::NAction (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.207.2.3 NCreate()

```
static bool gdcm::NormalizedNetworkFunctions::NCreate (
    const char * remote,
    uint16_t portno,
    BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.207.2.4 NDelete()

```
static bool gdcm::NormalizedNetworkFunctions::NDelete (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

### 10.207.2.5 NEventReport()

```
static bool gdcm::NormalizedNetworkFunctions::NEventReport (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

## 10.207.2.6 NGet()

```
static bool gdcm::NormalizedNetworkFunctions::NGet (
    const char * remote,
    uint16_t portno,
    const BaseQuery * query,
    std::vector< DataSet > & retDataSets,
    const char * aetitle,
    const char * call ) [static]
```

## 10.207.2.7 NSet()

```
static bool gdcm::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:

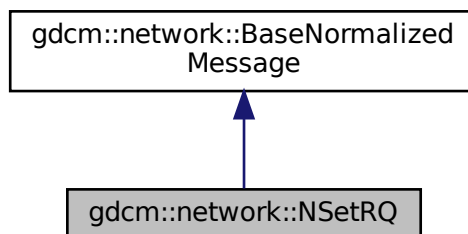
- [gdcmNormalizedNetworkFunctions.h](#)

## 10.208 gdcm::network::NSetRQ Class Reference

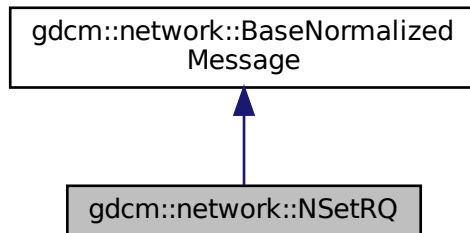
[NSetRQ](#).

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for gdcm::network::NSetRQ:



Collaboration diagram for `gdcm::network::NSetRQ`:



## Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &`inConnection`, const [BaseQuery](#) \*`inQuery`)

### 10.208.1 Detailed Description

[NSetRQ](#).

this file defines the messages for the nset action

### 10.208.2 Member Function Documentation

#### 10.208.2.1 ConstructPDV()

```
std::vector<PresentationDataValue> gdcm::network::NSetRQ::ConstructPDV (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [virtual]
```

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

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

- [gdcmNSetMessages.h](#)

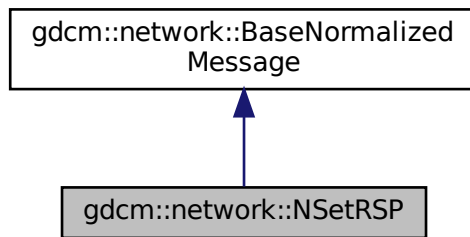


## 10.209 gdcm::network::NSetRSP Class Reference

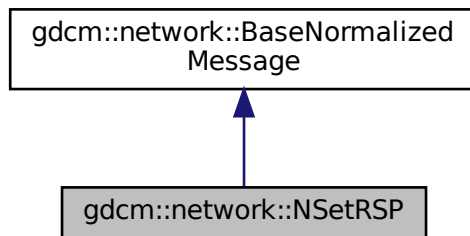
[NSetRSP](#) this file defines the messages for the nset action.

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for gdcm::network::NSetRSP:



Collaboration diagram for gdcm::network::NSetRSP:



### Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

### 10.209.1 Detailed Description

[NSetRSP](#) this file defines the messages for the nset action.

## 10.209.2 Member Function Documentation

### 10.209.2.1 ConstructPDVByDataSet()

```
std::vector<PresentationDataValue> gdcmm::network::NSetRSP::ConstructPDVByDataSet (
    const DataSet * inDataSet )
```

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

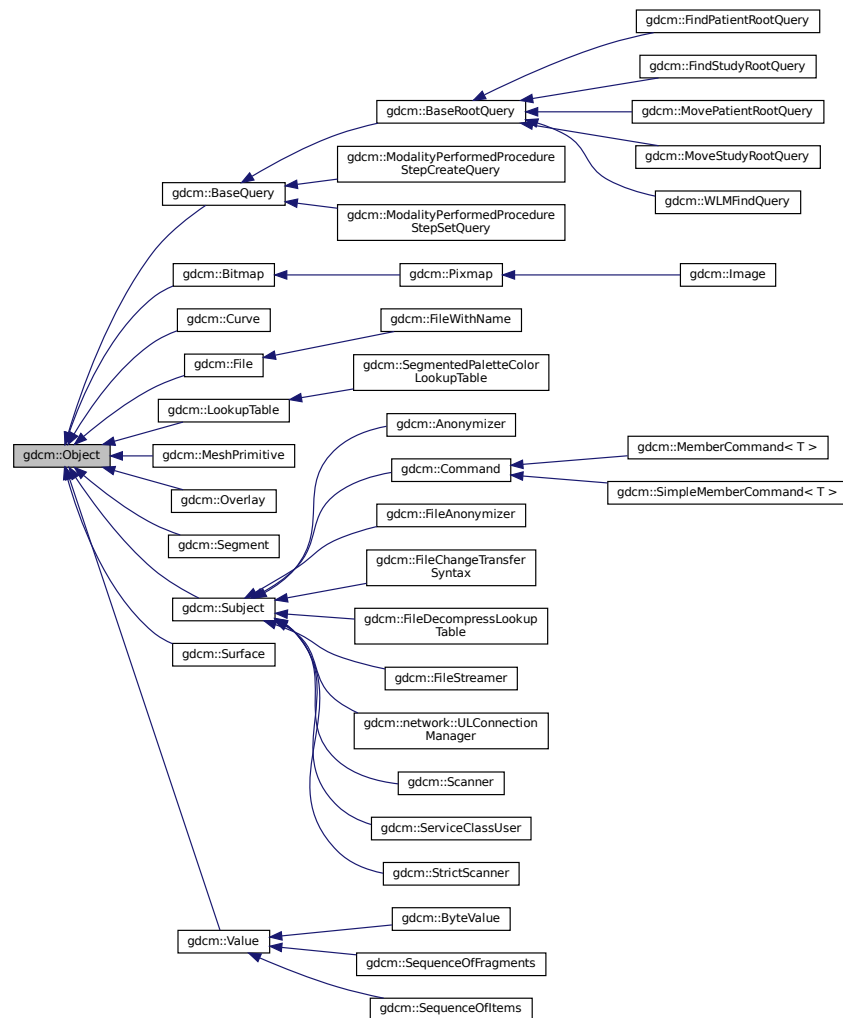
- [gdcmmNSetMessages.h](#)

## 10.210 gdcmm::Object Class Reference

[Object](#).

```
#include <gdcmmObject.h>
```

Inheritance diagram for gdcmm::Object:



## Public Member Functions

- [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

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

## Friends

- `std::ostream & operator<< (std::ostream &os, const Object &obj)`
- `template<class ObjectType >`  
`class SmartPointer`

### 10.210.1 Detailed Description

[Object](#).

#### Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

#### See also

[SmartPointer](#)

### 10.210.2 Constructor & Destructor Documentation

#### 10.210.2.1 [Object\(\)](#) [1/2]

```
gdcmm::Object::Object ( ) [inline]
```

#### 10.210.2.2 [~Object\(\)](#)

```
virtual gdcmm::Object::~~Object ( ) [inline], [virtual]
```

#### 10.210.2.3 [Object\(\)](#) [2/2]

```
gdcmm::Object::Object (
    const Object & ) [inline]
```

Special requirement for copy/cstor, assignment operator.

### 10.210.3 Member Function Documentation

#### 10.210.3.1 operator=()

```
void gdcm::Object::operator= (
    const Object & ) [inline]
```

#### 10.210.3.2 Print()

```
virtual void gdcm::Object::Print (
    std::ostream & ) const [inline], [virtual]
```

Reimplemented in [gdcm::SequenceOfFragments](#), [gdcm::SequenceOfItems](#), [gdcm::ByteValue](#), [gdcm::Scanner](#), [gdcm::StrictScanner](#), [gdcm::Image](#), [gdcm::BaseQuery](#), [gdcm::Curve](#), [gdcm::Overlay](#), [gdcm::Bitmap](#), [gdcm::LookupTable](#), [gdcm::Pixmap](#), and [gdcm::SegmentedPaletteColorLookupTable](#).

#### Examples:

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by [gdcm::operator<<\(\)](#).

#### 10.210.3.3 Register()

```
void gdcm::Object::Register ( ) [inline], [protected]
```

#### 10.210.3.4 UnRegister()

```
void gdcm::Object::UnRegister ( ) [inline], [protected]
```

### 10.210.4 Friends And Related Function Documentation

#### 10.210.4.1 operator<<

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

#### 10.210.4.2 SmartPointer

```
template<class ObjectType >  
friend class SmartPointer [friend]
```

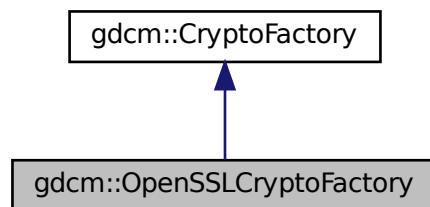
The documentation for this class was generated from the following file:

- [gdcmmObject.h](#)

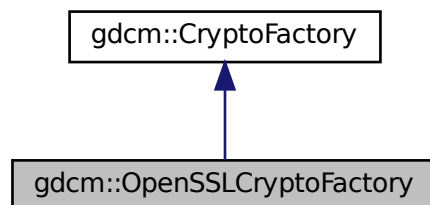
### 10.211 gdcmm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcmm::OpenSSLCryptoFactory:



Collaboration diagram for gdcmm::OpenSSLCryptoFactory:



## Public Member Functions

- [OpenSSLCryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Protected Member Functions

- void [InitOpenSSL](#) ()

## Additional Inherited Members

### 10.211.1 Constructor & Destructor Documentation

#### 10.211.1.1 OpenSSLCryptoFactory()

```
gdcmm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (
    CryptoLib id ) [inline]
```

References [gdcmmDebugMacro](#).

### 10.211.2 Member Function Documentation

#### 10.211.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcmm::OpenSSLCryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements [gdcmm::CryptoFactory](#).

#### 10.211.2.2 InitOpenSSL()

```
void gdcmm::OpenSSLCryptoFactory::InitOpenSSL ( ) [protected]
```

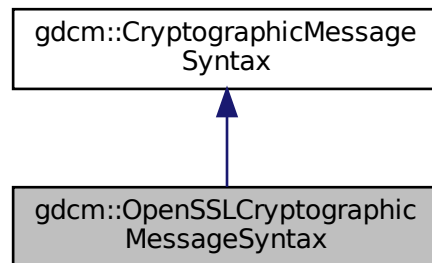
The documentation for this class was generated from the following file:

- [gdcmmOpenSSLCryptoFactory.h](#)

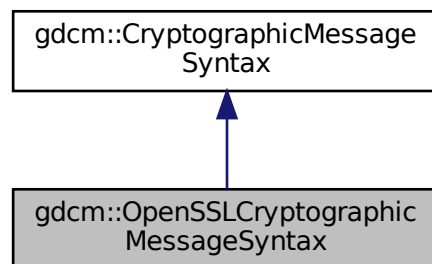
## 10.212 gdcM::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcMOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcM::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)

## Additional Inherited Members

### 10.212.1 Constructor & Destructor Documentation

#### 10.212.1.1 OpenSSLCryptographicMessageSyntax()

```
gdcmm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )
```

#### 10.212.1.2 ~OpenSSLCryptographicMessageSyntax()

```
gdcmm::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ( )
```

### 10.212.2 Member Function Documentation

#### 10.212.2.1 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](#).

#### 10.212.2.2 Encrypt()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::Encrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

create a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.212.2.3 GetCipherType()

```
CipherTypes gdcM::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.212.2.4 ParseCertificateFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.212.2.5 ParseKeyFile()

```
bool gdcM::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcM::CryptographicMessageSyntax](#).

#### 10.212.2.6 SetCipherType()

```
void gdcM::OpenSSLCryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcM::CryptographicMessageSyntax](#).

## 10.212.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
    const char * pass,
    size_t passLen ) [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

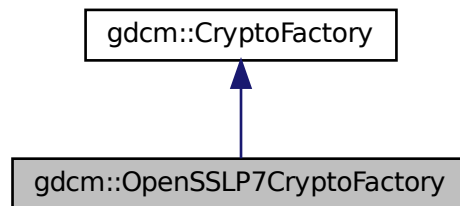
The documentation for this class was generated from the following file:

- [gdcmOpenSSLCryptographicMessageSyntax.h](#)

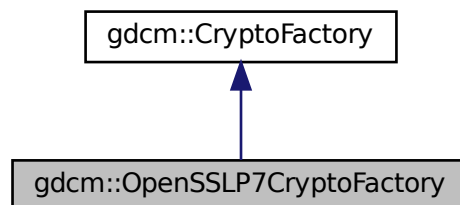
## 10.213 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



## Public Member Functions

- [OpenSSLP7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Additional Inherited Members

### 10.213.1 Constructor & Destructor Documentation

#### 10.213.1.1 OpenSSLP7CryptoFactory()

```
gdcmm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (  
    CryptoLib id ) [inline]
```

References [gdcmmDebugMacro](#).

### 10.213.2 Member Function Documentation

#### 10.213.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax* gdcmm::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]
```

Implements [gdcmm::CryptoFactory](#).

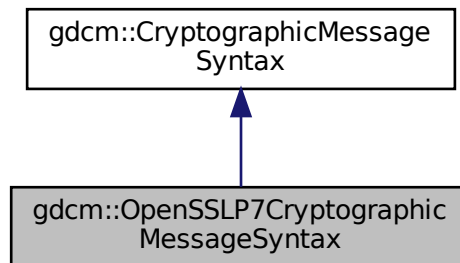
The documentation for this class was generated from the following file:

- [gdcmmOpenSSLP7CryptoFactory.h](#)

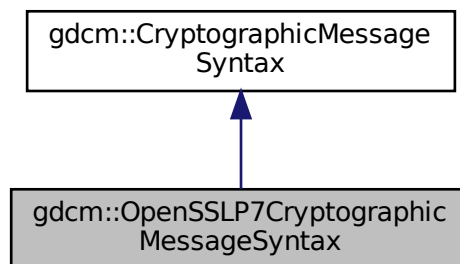
## 10.214 gdcM::OpenSSL7CryptographicMessageSyntax Class Reference

```
#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)

## Additional Inherited Members

### 10.214.1 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](http://www.openssl.org/docs/crypto/PKCS7_encrypt.html)

### 10.214.2 Constructor & Destructor Documentation

#### 10.214.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcM::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )
```

#### 10.214.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcM::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax ( )
```

### 10.214.3 Member Function Documentation

#### 10.214.3.1 Decrypt()

```
bool gdcM::OpenSSLP7CryptographicMessageSyntax::Decrypt (
    char * output,
    size_t & outlen,
    const char * array,
    size_t len ) const [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

### 10.214.3.2 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](#).

### 10.214.3.3 GetCipherType()

```
CipherTypes gdcmm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.214.3.4 ParseCertificateFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.214.3.5 ParseKeyFile()

```
bool gdcmm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
    const char * filename ) [virtual]
```

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.214.3.6 SetCipherType()

```
void gdcmm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
    CipherTypes type ) [virtual]
```

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

### 10.214.3.7 SetPassword()

```
bool gdcm::OpenSSL7CryptographicMessageSyntax::SetPassword (
    const char * ,
    size_t ) [inline], [virtual]
```

Implements [gdcm::CryptographicMessageSyntax](#).

References [gdcmWarningMacro](#).

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

- [gdcmOpenSSL7CryptographicMessageSyntax.h](#)

## 10.215 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](#) ()
- 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)

### 10.215.1 Detailed Description

class to handle [Orientation](#)

### 10.215.2 Member Enumeration Documentation

#### 10.215.2.1 OrientationType

enum [gdcm::Orientation::OrientationType](#)

##### Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

### 10.215.3 Constructor & Destructor Documentation

#### 10.215.3.1 Orientation()

[gdcm::Orientation::Orientation](#) ( )

#### 10.215.3.2 ~Orientation()

[gdcm::Orientation::~~Orientation](#) ( )

## 10.215.4 Member Function Documentation

### 10.215.4.1 GetLabel()

```
static const char* gdcm::Orientation::GetLabel (
    OrientationType type ) [static]
```

Return the label of an [Orientation](#).

Examples:

[FixOrientation.cxx](#).

### 10.215.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
    double x,
    double y,
    double z ) [static], [protected]
```

### 10.215.4.3 GetObliquityThresholdCosineValue()

```
static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]
```

### 10.215.4.4 GetType()

```
static 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](#).

#### 10.215.4.5 Print()

```
void gdcm::Orientation::Print (
    std::ostream & ) const
```

Print.

Referenced by `gdcm::operator<<()`.

#### 10.215.4.6 SetObliquityThresholdCosineValue()

```
static void gdcm::Orientation::SetObliquityThresholdCosineValue (
    double val ) [static]
```

ObliquityThresholdCosineValue stuff.

### 10.215.5 Friends And Related Function Documentation

#### 10.215.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Orientation & o ) [friend]
```

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

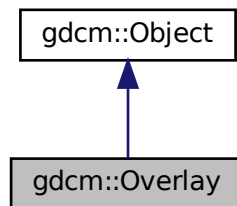
- [gdcmOrientation.h](#)

## 10.216 gdcM::Overlay Class Reference

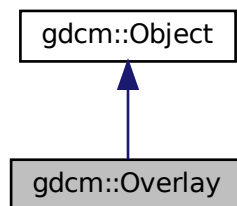
[Overlay](#) class.

```
#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](#) ()
- 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  
*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:*

### Static Public Member Functions

- static const char \* [GetOverlayTypeAsString](#) ([OverlayType](#) ot)
- static [OverlayType](#) [GetOverlayTypeFromString](#) (const char \*)

### Additional Inherited Members

#### 10.216.1 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:

#### 10.216.2 Member Enumeration Documentation

##### 10.216.2.1 OverlayType

enum [gdcm::Overlay::OverlayType](#)

**Enumerator**

Invalid	
Graphics	
ROI	

**10.216.3 Constructor & Destructor Documentation****10.216.3.1 Overlay()** [1/2]

```
gdcm::Overlay::Overlay ( )
```

**10.216.3.2 ~Overlay()**

```
gdcm::Overlay::~~Overlay ( )
```

**10.216.3.3 Overlay()** [2/2]

```
gdcm::Overlay::Overlay (
    Overlay const & ov )
```

**10.216.4 Member Function Documentation****10.216.4.1 Decompress()**

```
void gdcm::Overlay::Decompress (
    std::ostream & os ) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

**10.216.4.2 GetBitPosition()**

```
unsigned short gdcm::Overlay::GetBitPosition ( ) const
```

return bit position

**10.216.4.3 GetBitsAllocated()**

```
unsigned short gdcm::Overlay::GetBitsAllocated ( ) const
```

return bits allocated

**10.216.4.4 GetColumns()**

```
unsigned short gdcm::Overlay::GetColumns ( ) const
```

get columns

**10.216.4.5 GetDescription()**

```
const char* gdcm::Overlay::GetDescription ( ) const
```

get description

**10.216.4.6 GetGroup()**

```
unsigned short gdcm::Overlay::GetGroup ( ) const
```

Get Group number.

**10.216.4.7 GetOrigin()**

```
const signed short* gdcm::Overlay::GetOrigin ( ) const
```

get origin



#### 10.216.4.8 GetOverlayData()

```
const ByteValue& gdcm::Overlay::GetOverlayData ( ) const
```

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

#### 10.216.4.9 GetOverlayTypeAsString()

```
static const char* gdcm::Overlay::GetOverlayTypeAsString (
    OverlayType ot ) [static]
```

#### 10.216.4.10 GetOverlayTypeFromString()

```
static OverlayType gdcm::Overlay::GetOverlayTypeFromString (
    const char * ) [static]
```

#### 10.216.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows ( ) const
```

get rows

#### 10.216.4.12 GetType()

```
const char* gdcm::Overlay::GetType ( ) const
```

get type

#### 10.216.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const
```

**10.216.4.14 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\(\)](#)

**10.216.4.15 GetUnpackBufferLength()**

```
size_t gdcM::Overlay::GetUnpackBufferLength ( ) const
```

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

**10.216.4.16 GrabOverlayFromPixelData()**

```
bool gdcM::Overlay::GrabOverlayFromPixelData (
    DataSet const & ds )
```

**10.216.4.17 IsEmpty()**

```
bool gdcM::Overlay::IsEmpty ( ) const
```

Return whether or not the [Overlay](#) is empty:

**10.216.4.18 IsInPixelData()** [1/2]

```
bool gdcM::Overlay::IsInPixelData ( ) const
```

return if the [Overlay](#) is stored in the pixel data or not

**10.216.4.19 IsInPixelData()** [2/2]

```
void gdcM::Overlay::IsInPixelData (
    bool b )
```

Set whether or no the OverlayData is in the Pixel Data:

#### 10.216.4.20 IsZero()

```
bool gdcm::Overlay::IsZero ( ) const
```

return true if all bits are set to 0

#### 10.216.4.21 operator=()

```
Overlay& gdcm::Overlay::operator= (
    Overlay const & ov )
```

#### 10.216.4.22 Print()

```
void gdcm::Overlay::Print (
    std::ostream & ) const [virtual]
```

Print.

Reimplemented from [gdcm::Object](#).

#### 10.216.4.23 SetBitPosition()

```
void gdcm::Overlay::SetBitPosition (
    unsigned short bitposition )
```

set bit position

#### 10.216.4.24 SetBitsAllocated()

```
void gdcm::Overlay::SetBitsAllocated (
    unsigned short bitsallocated )
```

set bits allocated

**10.216.4.25 SetColumns()**

```
void gdcM::Overlay::SetColumns (
    unsigned short columns )
```

set columns

**10.216.4.26 SetDescription()**

```
void gdcM::Overlay::SetDescription (
    const char * description )
```

set description

**10.216.4.27 SetFrameOrigin()**

```
void gdcM::Overlay::SetFrameOrigin (
    unsigned short frameorigin )
```

set frame origin

**10.216.4.28 SetGroup()**

```
void gdcM::Overlay::SetGroup (
    unsigned short group )
```

Set Group number.

**10.216.4.29 SetNumberOfFrames()**

```
void gdcM::Overlay::SetNumberOfFrames (
    unsigned int numberofframes )
```

set number of frames

#### 10.216.4.30 SetOrigin()

```
void gdcmm::Overlay::SetOrigin (
    const signed short origin[2] )
```

set origin

#### 10.216.4.31 SetOverlay()

```
void gdcmm::Overlay::SetOverlay (
    const char * array,
    size_t length )
```

set overlay from byte array + length

#### 10.216.4.32 SetRows()

```
void gdcmm::Overlay::SetRows (
    unsigned short rows )
```

set rows

#### 10.216.4.33 SetType()

```
void gdcmm::Overlay::SetType (
    const char * type )
```

set type

#### 10.216.4.34 Update()

```
void gdcmm::Overlay::Update (
    const DataElement & de )
```

Update overlay from data element de:

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

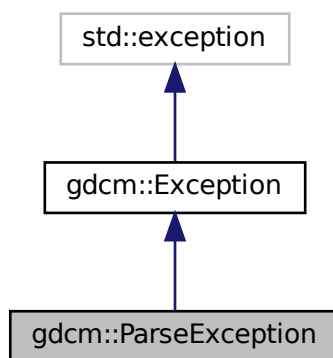
- [gdcmmOverlay.h](#)

## 10.217 gdcm::ParseException Class Reference

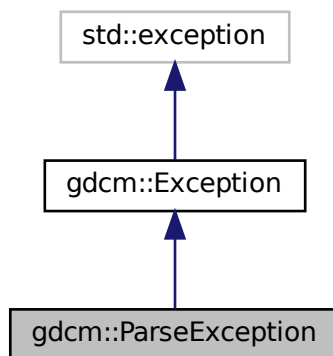
[ParseException](#) Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



## Public Member Functions

- [ParseException](#) ()
- virtual [~ParseException](#) () throw ()
- const [DataElement](#) & [GetLastElement](#) () const
- [ParseException](#) & [operator=](#) (const [ParseException](#) &orig)
- void [SetLastElement](#) ([DataElement](#) &de)

### 10.217.1 Detailed Description

[ParseException](#) Standard exception handling object.

### 10.217.2 Constructor & Destructor Documentation

#### 10.217.2.1 ParseException()

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

#### 10.217.2.2 ~ParseException()

```
virtual gdcm::ParseException::~~ParseException ( ) throw ( ) [inline], [virtual]
```

### 10.217.3 Member Function Documentation

#### 10.217.3.1 GetLastElement()

```
const DataElement& gdcm::ParseException::GetLastElement ( ) const [inline]
```

#### 10.217.3.2 operator=()

```
ParseException& gdcm::ParseException::operator= (
    const ParseException & orig ) [inline]
```

Assignment operator.

### 10.217.3.3 SetLastElement()

```
void gdcM::ParseException::SetLastElement (
    DataElement & de ) [inline]
```

Equivalence operator.

Referenced by gdcM::Fragment::ReadBacktrack(), and gdcM::Fragment::ReadValue().

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

- [gdcMParseException.h](#)

## 10.218 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](#) ()

### 10.218.1 Detailed Description

[Parser](#) ala XML\_Parser from expat (SAX)

Detailled description here

#### Note

Simple API for DICOM

### 10.218.2 Member Typedef Documentation

#### 10.218.2.1 EndElementHandler

```
typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)
```

#### 10.218.2.2 StartElementHandler

```
typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char  
*atts[])
```

### 10.218.3 Member Enumeration Documentation

#### 10.218.3.1 ErrorType

```
enum gdcm::Parser::ErrorType
```

**Enumerator**

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

**10.218.4 Constructor & Destructor Documentation****10.218.4.1 Parser()**

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

**10.218.4.2 ~Parser()**

```
gdcm::Parser::~~Parser ( ) [inline]
```

**10.218.5 Member Function Documentation****10.218.5.1 GetBuffer()**

```
char* gdcm::Parser::GetBuffer (
    int len ) [protected]
```

**10.218.5.2 GetCurrentByteIndex()**

```
unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const
```

### 10.218.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode ( ) const
```

### 10.218.5.4 GetErrorString()

```
static const char* gdcm::Parser::GetErrorString (
    ErrorType const & err ) [static]
```

### 10.218.5.5 GetUserData()

```
void* gdcm::Parser::GetUserData ( ) const
```

### 10.218.5.6 Parse()

```
bool gdcm::Parser::Parse (
    const char * s,
    int len,
    bool isFinal )
```

### 10.218.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
    int len,
    bool isFinal ) [protected]
```

### 10.218.5.8 Process()

```
ErrorType gdcm::Parser::Process ( ) [protected]
```

#### 10.218.5.9 SetElementHandler()

```
void gdcM::Parser::SetElementHandler (
    StartElementHandler start,
    EndElementHandler end )
```

#### 10.218.5.10 SetUserData()

```
void gdcM::Parser::SetUserData (
    void * userData )
```

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

- [gdcMParser.h](#)

### 10.219 gdcM::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcMPatient.h>
```

#### Public Member Functions

- [Patient](#) ()

#### 10.219.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

#### 10.219.2 Constructor & Destructor Documentation

##### 10.219.2.1 Patient()

```
gdcM::Patient::Patient ( ) [inline]
```

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

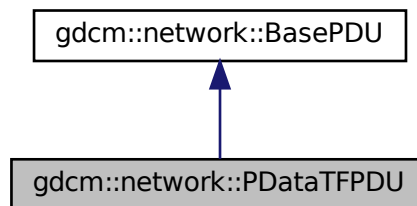
- [gdcMPatient.h](#)

## 10.220 gdcm::network::PDataTFPDU Class Reference

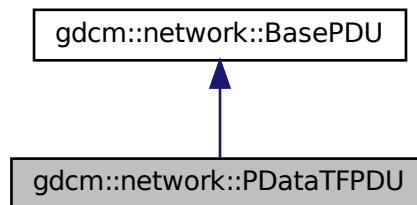
[PDataTFPDU](#).

```
#include <gdcmPDataTFPDU.h>
```

Inheritance diagram for gdcm::network::PDataTFPDU:



Collaboration diagram for gdcm::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
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

## Protected Member Functions

- `std::istream & ReadInto (std::istream &is, std::ostream &os)`

### 10.220.1 Detailed Description

[PDataTFPDU](#).

[Table 9-22](#) P-DATA-TF PDU FIELDS

### 10.220.2 Member Typedef Documentation

#### 10.220.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcn::network::PDataTFPDU::SizeType
```

### 10.220.3 Constructor & Destructor Documentation

#### 10.220.3.1 PDataTFPDU()

```
gdcn::network::PDataTFPDU::PDataTFPDU ( )
```

### 10.220.4 Member Function Documentation

#### 10.220.4.1 AddPresentationDataValue()

```
void gdcn::network::PDataTFPDU::AddPresentationDataValue (  
    PresentationDataValue const & pdv ) [inline]
```

#### 10.220.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]
```

#### 10.220.4.3 GetPresentationDataValue()

```
PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (
    SizeType i ) const [inline]
```

#### 10.220.4.4 IsLastFragment()

```
bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const [virtual]
```

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

#### 10.220.4.5 Print()

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

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

#### 10.220.4.6 Read()

```
std::istream& gdcm::network::PDataTFPDU::Read (
    std::istream & is ) [virtual]
```

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

#### 10.220.4.7 ReadInto()

```
std::istream& gdcm::network::PDataTFPDU::ReadInto (
    std::istream & is,
    std::ostream & os ) [protected]
```

#### 10.220.4.8 Size()

```
size_t gdcn::network::PDataTFPDU::Size ( ) const [virtual]
```

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

#### 10.220.4.9 Write()

```
const std::ostream& gdcn::network::PDataTFPDU::Write (
    std::ostream & os ) const [virtual]
```

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

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

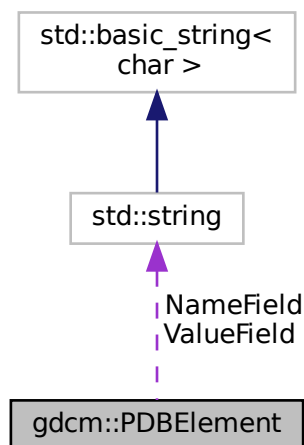
- [gdcnPDataTFPDU.h](#)

## 10.221 gdcn::PDBelement Class Reference

Class to represent a PDB [Element](#).

```
#include <gdcnPDBelement.h>
```

Collaboration diagram for gdcn::PDBelement:





## Public Member Functions

- [PDBelement](#) ()
- 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)

### 10.221.1 Detailed Description

Class to represent a PDB [Element](#).

See also

[PDBHeader](#)

### 10.221.2 Constructor & Destructor Documentation

#### 10.221.2.1 PDBelement()

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

References [gdcm::operator<<\(\)](#).

### 10.221.3 Member Function Documentation

#### 10.221.3.1 GetName()

```
const char* gdcm::PDBelement::GetName ( ) const [inline]
```

Set/Get Name.

#### 10.221.3.2 GetValue()

```
const char* gdcm::PDBelement::GetValue ( ) const [inline]
```

Set/Get [Value](#).

#### 10.221.3.3 operator==( )

```
bool gdcm::PDBelement::operator== (
    const PDBelement & de ) const [inline]
```

References [NameField](#), and [ValueField](#).

#### 10.221.3.4 SetName()

```
void gdcm::PDBelement::SetName (
    const char * name ) [inline]
```

#### 10.221.3.5 SetValue()

```
void gdcm::PDBelement::SetValue (
    const char * value ) [inline]
```

### 10.221.4 Friends And Related Function Documentation

## 10.221.4.1 operator&lt;&lt;

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

## 10.221.5 Member Data Documentation

## 10.221.5.1 NameField

```
std::string gdcm::PDBElement::NameField [protected]
```

Referenced by `gdcm::operator<<()`, and `operator==()`.

## 10.221.5.2 ValueField

```
std::string gdcm::PDBElement::ValueField [protected]
```

Referenced by `gdcm::operator<<()`, and `operator==()`.

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

- [gdcmPDBElement.h](#)

## 10.222 gdcm::PDBHeader Class Reference

Class for [PDBHeader](#).

```
#include <gdcmPDBHeader.h>
```

## Public Member Functions

- [PDBHeader](#) ()
- [~PDBHeader](#) ()
- 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)

### 10.222.1 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.

#### See also

[CSAHeader](#)

### 10.222.2 Constructor & Destructor Documentation

#### 10.222.2.1 PDBHeader()

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

#### 10.222.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~~PDBHeader ( ) [inline]
```

### 10.222.3 Member Function Documentation

#### 10.222.3.1 FindPDBElementByName()

```
bool gdcm::PDBHeader::FindPDBElementByName (
    const char * name )
```

Return true if the PDB element matching name is found or not.

#### 10.222.3.2 GetPDBEEnd()

```
const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const [protected]
```

#### 10.222.3.3 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

#### 10.222.3.4 GetPDBInfoTag()

```
static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( ) [static]
```

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

### 10.222.3.5 LoadFromDataElement()

```
bool gdcM::PDBHeader::LoadFromDataElement (
    DataElement const & de )
```

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

### 10.222.3.6 Print()

```
void gdcM::PDBHeader::Print (
    std::ostream & os ) const
```

Print.

Referenced by `gdcM::operator<<()`.

## 10.222.4 Friends And Related Function Documentation

### 10.222.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const PDBHeader & d ) [friend]
```

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

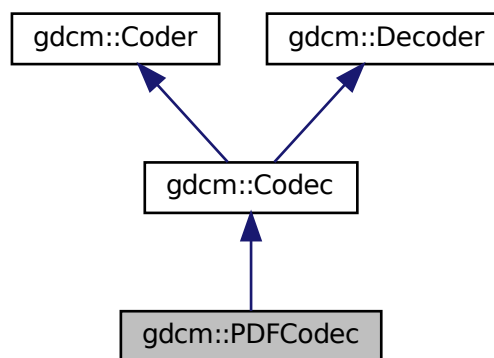
- [gdcMPDBHeader.h](#)

## 10.223 gdcm::PDFCodec Class Reference

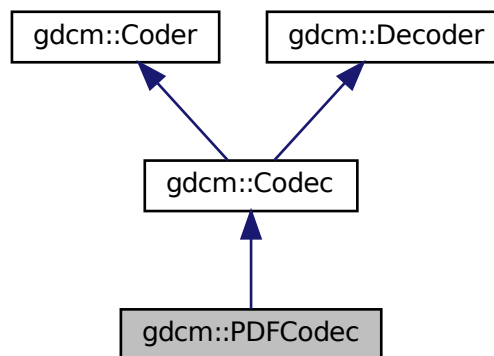
[PDFCodec](#) class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



## Public Member Functions

- [PDFCodec](#) ()
- [~PDFCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &) const  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &) const  
*Return whether this decoder support this transfer syntax (can decode it)*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*

## Additional Inherited Members

### 10.223.1 Detailed Description

[PDFCodec](#) class.

### 10.223.2 Constructor & Destructor Documentation

#### 10.223.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ( )
```

#### 10.223.2.2 ~PDFCodec()

```
gdcm::PDFCodec::~~PDFCodec ( )
```

### 10.223.3 Member Function Documentation

#### 10.223.3.1 CanCode()

```
bool gdcm::PDFCodec::CanCode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).



## 10.223.3.2 CanDecode()

```
bool gdcm::PDFCodec::CanDecode (
    TransferSyntax const & ) const [inline], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

## 10.223.3.3 Decode()

```
bool gdcm::PDFCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::Decoder](#).

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

- [gdcmPDFCodec.h](#)

## 10.224 gdcm::network::PDUFactory Class Reference

[PDUFactory](#) basically, given an initial byte, construct the.

```
#include <gdcmPDUFactory.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](#) \*> &inDataPDUs)

### 10.224.1 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.

### 10.224.2 Member Function Documentation

#### 10.224.2.1 ConstructAbortPDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructAbortPDU ( ) [static]
```

#### 10.224.2.2 ConstructPDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructPDU (
    uint8_t itemtype ) [static]
```

#### 10.224.2.3 ConstructReleasePDU()

```
static BasePDU* gdcn::network::PDUFactory::ConstructReleasePDU ( ) [static]
```

#### 10.224.2.4 CreateCEchoPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCEchoPDU (
    const ULConnection & inConnection ) [static]
```

#### 10.224.2.5 CreateCFindPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCFindPDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

#### 10.224.2.6 CreateCMovePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCMovePDU (
    const ULConnection & inConnection,
    const BaseRootQuery * inRootQuery ) [static]
```

#### 10.224.2.7 CreateCStoreRQPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRQPDU (
    const ULConnection & inConnection,
    const File & file,
    bool writeDataSet = true ) [static]
```

#### 10.224.2.8 CreateCStoreRSPPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRSPPDU (
    const DataSet * inDataSet,
    const BasePDU * inPC ) [static]
```

#### 10.224.2.9 CreateNActionPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNActionPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.10 CreateNCreatePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNCreatePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.11 CreateNDeletePDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNDeletePDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.12 CreateNEventReportPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNEventReportPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.13 CreateNGetPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNGetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.14 CreateNSetPDU()

```
static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateNSetPDU (
    const ULConnection & inConnection,
    const BaseQuery * inQuery ) [static]
```

#### 10.224.2.15 DetermineEventByPDU()

```
static EEventID gdcm::network::PDUFactory::DetermineEventByPDU (
    const BasePDU * inPDU ) [static]
```

### 10.224.2.16 GetPDVs()

```
static std::vector<PresentationDataValue> gdcm::network::PDUFactory::GetPDVs (
    const std::vector< BasePDU *> & inDataPDUs ) [static]
```

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

- [gdcmPDUFactory.h](#)

## 10.225 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](#) = '^'

### 10.225.1 Detailed Description

[PersonName](#) class.

### 10.225.2 Member Function Documentation

#### 10.225.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength ( ) const [inline]
```

#### 10.225.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const [inline]
```

#### 10.225.2.3 Print()

```
void gdcm::PersonName::Print (
    std::ostream & os ) const [inline]
```

#### 10.225.2.4 SetBlob()

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

#### 10.225.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
    const char * comp1 = "",
    const char * comp2 = "",
    const char * comp3 = "",
    const char * comp4 = "",
    const char * comp5 = "" ) [inline]
```

#### 10.225.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (
    const char * components[] ) [inline]
```

### 10.225.3 Member Data Documentation

#### 10.225.3.1 Component

```
char gdcm::PersonName::Component [MaxNumberOfComponents] [MaxLength+1]
```

#### 10.225.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64 [static]
```

#### 10.225.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 [static]
```

#### 10.225.3.4 Padding

```
const char gdcm::PersonName::Padding = ' ' [static]
```

#### 10.225.3.5 Separator

```
const char gdcm::PersonName::Separator = '^' [static]
```

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

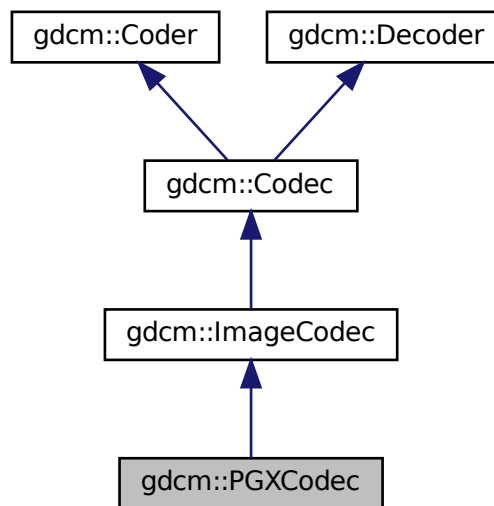
- [gdcmPersonName.h](#)

## 10.226 gdcm::PGXCodec Class Reference

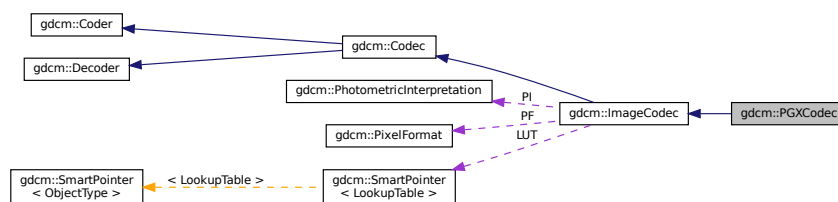
Class to do PGX.

```
#include <gdcmPGXCodec.h>
```

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



### Public Member Functions

- [PGXCodec](#) ()
- [~PGXCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const



*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

*Return whether this decoder support this transfer syntax (can decode it)*

- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [Read](#) (const char \*filename, [DataElement](#) &out) const
- bool [Write](#) (const char \*filename, const [DataElement](#) &out) const

## Additional Inherited Members

### 10.226.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

### 10.226.2 Constructor & Destructor Documentation

#### 10.226.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ( )
```

#### 10.226.2.2 ~PGXCodec()

```
gdcm::PGXCodec::~~PGXCodec ( )
```

### 10.226.3 Member Function Documentation

#### 10.226.3.1 CanCode()

```
bool gdcm::PGXCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.226.3.2 CanDecode()

```
bool gdcM::PGXCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcM::ImageCodec](#).

### 10.226.3.3 Clone()

```
virtual ImageCodec* gdcM::PGXCodec::Clone ( ) const [virtual]
```

Implements [gdcM::ImageCodec](#).

### 10.226.3.4 GetHeaderInfo()

```
bool gdcM::PGXCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcM::ImageCodec](#).

### 10.226.3.5 Read()

```
bool gdcM::PGXCodec::Read (
    const char * filename,
    DataElement & out ) const
```

### 10.226.3.6 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](#)

## 10.227 gdcm::PhotometricInterpretation Class Reference

Class to represent an [PhotometricInterpretation](#).

```
#include <gdcmPhotometricInterpretation.h>
```

### Public Types

- enum [PType](#) {  
    [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](#) ([PType](#) pi=[UNKNOWN](#))
- unsigned short [GetSamplesPerPixel](#) () const  
*return the value for Sample Per Pixel associated with a particular Photometric Interpretation*
- const char \* [GetString](#) () const
- [PType](#) [GetType](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsSameColorSpace](#) ([PhotometricInterpretation](#) const &pi) const
- [operator PType](#) () const

### Static Public Member Functions

- static const char \* [GetPIString](#) ([PType](#) pi)
- static [PType](#) [GetPType](#) (const char \*pi)
- static bool [IsRetired](#) ([PType](#) pi)

### Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &pi)

### 10.227.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [HelloVizWorld.cxx](#), and [iU22tomultisc.cxx](#).

### 10.227.2 Member Enumeration Documentation

#### 10.227.2.1 PType

```
enum gdcm::PhotometricInterpretation::PType
```

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	

### 10.227.3 Constructor & Destructor Documentation

#### 10.227.3.1 PhotometricInterpretation()

```
gdcm::PhotometricInterpretation::PhotometricInterpretation (
    PType pi = UNKNOWN ) [inline]
```

References [gdcm::operator<<\(\)](#).

## 10.227.4 Member Function Documentation

### 10.227.4.1 GetPIString()

```
static const char* gdcm::PhotometricInterpretation::GetPIString (
    PType pi ) [static]
```

Referenced by `gdcm::operator<<()`.

### 10.227.4.2 GetPType()

```
static PType gdcm::PhotometricInterpretation::GetPType (
    const char * pi ) [static]
```

### 10.227.4.3 GetSamplesPerPixel()

```
unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

### 10.227.4.4 GetString()

```
const char* gdcm::PhotometricInterpretation::GetString ( ) const
```

### 10.227.4.5 GetType()

```
PType gdcm::PhotometricInterpretation::GetType ( ) const [inline]
```

### 10.227.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless ( ) const
```

#### 10.227.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy ( ) const
```

#### 10.227.4.8 IsRetired()

```
static bool gdcm::PhotometricInterpretation::IsRetired (
    PType pi ) [static]
```

#### 10.227.4.9 IsSameColorSpace()

```
bool gdcm::PhotometricInterpretation::IsSameColorSpace (
    PhotometricInterpretation const & pi ) const
```

#### 10.227.4.10 operator PType()

```
gdcm::PhotometricInterpretation::operator PType ( ) const [inline]
```

### 10.227.5 Friends And Related Function Documentation

#### 10.227.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const PhotometricInterpretation & pi ) [friend]
```

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

- [gdcmPhotometricInterpretation.h](#)

## 10.228 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](#) (unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- [PixelFormat](#) ([ScalarType](#) st)
- 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!= ([ScalarType](#) st) const
- bool operator!= (const [PixelFormat](#) &pf) const
- bool operator== ([ScalarType](#) st) const
- bool operator== (const [PixelFormat](#) &pf) 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)

## 10.228.1 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](#), [csa2img.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSample←Precision.cxx](#), [iU22tomultisc.cxx](#), [TemplateEmptyImage.cxx](#), and [threadgdcm.cxx](#).

## 10.228.2 Member Enumeration Documentation

### 10.228.2.1 ScalarType

```
enum gdcm::PixelFormat::ScalarType
```



## Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	
INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

## 10.228.3 Constructor &amp; Destructor Documentation

## 10.228.3.1 PixelFormat() [1/2]

```
gdcm::PixelFormat::PixelFormat (
    unsigned short samplesperpixel = 1,
    unsigned short bitsallocated = 8,
    unsigned short bitsstored = 8,
    unsigned short highbit = 7,
    unsigned short pixelrepresentation = 0 ) [inline], [explicit]
```

## 10.228.3.2 PixelFormat() [2/2]

```
gdcm::PixelFormat::PixelFormat (
    ScalarType st )
```

## 10.228.4 Member Function Documentation

#### 10.228.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const [inline]
```

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

Examples:

[GetJPEGSamplePrecision.cxx](#).

#### 10.228.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]
```

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

Examples:

[GetJPEGSamplePrecision.cxx](#).

#### 10.228.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]
```

HighBit see [Tag](#) (0028,0102) US High Bit.

#### 10.228.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax ( ) const
```

return the max possible of the pixel

#### 10.228.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin ( ) const
```

return the min possible of the pixel

#### 10.228.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]
```

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

#### 10.228.4.7 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:

[threadgdcm.cxx](#).

#### 10.228.4.8 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](#).

#### 10.228.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType ( ) const
```

ScalarType does not take into account the sample per pixel.

**10.228.4.10 GetScalarTypeAsString()**

```
const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const
```

**10.228.4.11 IsCompatible()**

```
bool gdcm::PixelFormat::IsCompatible (
    const TransferSyntax & ts ) const
```

**10.228.4.12 IsValid()**

```
bool gdcm::PixelFormat::IsValid ( ) const
```

```
return IsValid
```

**10.228.4.13 operator ScalarType()**

```
gdcm::PixelFormat::operator ScalarType ( ) const [inline]
```

**10.228.4.14 operator!=( ) [1/2]**

```
bool gdcm::PixelFormat::operator!= (
    ScalarType st ) const [inline]
```

**10.228.4.15 operator!=( ) [2/2]**

```
bool gdcm::PixelFormat::operator!= (
    const PixelFormat & pf ) const [inline]
```

**10.228.4.16 operator==(** [1/2]

```
bool gdcm::PixelFormat::operator== (
    ScalarType st ) const [inline]
```

**10.228.4.17 operator==(** [2/2]

```
bool gdcm::PixelFormat::operator== (
    const PixelFormat & pf ) const [inline]
```

**10.228.4.18 Print()**

```
void gdcm::PixelFormat::Print (
    std::ostream & os ) const
```

Print.

Referenced by gdcm::operator<<().

**10.228.4.19 SetBitsAllocated()**

```
void gdcm::PixelFormat::SetBitsAllocated (
    unsigned short ba ) [inline]
```

**10.228.4.20 SetBitsStored()**

```
void gdcm::PixelFormat::SetBitsStored (
    unsigned short bs ) [inline]
```

**10.228.4.21 SetHighBit()**

```
void gdcm::PixelFormat::SetHighBit (
    unsigned short hb ) [inline]
```

#### 10.228.4.22 SetPixelRepresentation()

```
void gdcm::PixelFormat::SetPixelRepresentation (
    unsigned short pr ) [inline]
```

##### Examples:

[TemplateEmptyImage.cxx](#).

#### 10.228.4.23 SetSamplesPerPixel()

```
void gdcm::PixelFormat::SetSamplesPerPixel (
    unsigned short spp ) [inline]
```

##### Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [GenFakelImage.cxx](#).

References `gdcmAssertMacro`.

#### 10.228.4.24 SetScalarType()

```
void gdcm::PixelFormat::SetScalarType (
    ScalarType st )
```

Set [PixelFormat](#) based only on the `ScalarType`

##### Warning

: You need to call `SetScalarType` *before* `SetSamplesPerPixel`

#### 10.228.4.25 Validate()

```
bool gdcm::PixelFormat::Validate ( ) [protected]
```

When image with 24/24/23 was read, need to validate.

Referenced by `gdcm::Bitmap::SetPixelFormat()`.

## 10.228.5 Friends And Related Function Documentation

### 10.228.5.1 Bitmap

```
friend class Bitmap [friend]
```

### 10.228.5.2 operator<<

```
std::ostream& operator<< (  
    std::ostream & _os,  
    const PixelFormat & pf ) [friend]
```

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

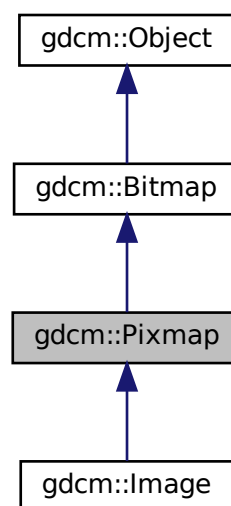
- [gdcmPixelFormat.h](#)

## 10.229 gdcm::Pixmap Class Reference

[Pixmap](#) class.

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:







## Additional Inherited Members

### 10.229.1 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](#)

### 10.229.2 Constructor & Destructor Documentation

#### 10.229.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ( )
```

#### 10.229.2.2 ~Pixmap()

```
gdcm::Pixmap::~~Pixmap ( )
```

### 10.229.3 Member Function Documentation

#### 10.229.3.1 AreOverlaysInPixelData()

```
bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const [virtual]
```

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

**10.229.3.2 GetCurve()** [1/2]

```
Curve& gdcm::Pixmap::GetCurve (
    size_t i = 0 ) [inline]
```

Curve: group 50xx.

**10.229.3.3 GetCurve()** [2/2]

```
const Curve& gdcm::Pixmap::GetCurve (
    size_t i = 0 ) const [inline]
```

**10.229.3.4 GetIconImage()** [1/2]

```
const IconImage& gdcm::Pixmap::GetIconImage ( ) const [inline]
```

Set/Get Icon Image.

**10.229.3.5 GetIconImage()** [2/2]

```
IconImage& gdcm::Pixmap::GetIconImage ( ) [inline]
```

**10.229.3.6 GetNumberOfCurves()**

```
size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]
```

**10.229.3.7 GetNumberOfOverlays()**

```
size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]
```

**10.229.3.8 GetOverlay()** [1/2]

```
Overlay& gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) [inline]
```

Overlay: group 60xx.

**10.229.3.9 GetOverlay()** [2/2]

```
const Overlay& gdcm::Pixmap::GetOverlay (
    size_t i = 0 ) const [inline]
```

**10.229.3.10 Print()**

```
void gdcm::Pixmap::Print (
    std::ostream & ) const [virtual]
```

Reimplemented from [gdcm::Bitmap](#).

**10.229.3.11 RemoveOverlay()**

```
void gdcm::Pixmap::RemoveOverlay (
    size_t i ) [inline]
```

**10.229.3.12 SetIconImage()**

```
void gdcm::Pixmap::SetIconImage (
    IconImage const & ii ) [inline]
```

**10.229.3.13 SetNumberOfCurves()**

```
void gdcm::Pixmap::SetNumberOfCurves (
    size_t n ) [inline]
```

#### 10.229.3.14 SetNumberOfOverlays()

```
void gdcM::Pixmap::SetNumberOfOverlays (
    size_t n ) [inline]
```

#### 10.229.3.15 UnusedBitsPresentInPixelData()

```
bool gdcM::Pixmap::UnusedBitsPresentInPixelData ( ) const [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from [gdcM::Bitmap](#).

### 10.229.4 Member Data Documentation

#### 10.229.4.1 Curves

```
std::vector<Curve> gdcM::Pixmap::Curves [protected]
```

#### 10.229.4.2 Icon

```
SmartPointer<IconImage> gdcM::Pixmap::Icon [protected]
```

#### 10.229.4.3 Overlays

```
std::vector<Overlay> gdcM::Pixmap::Overlays [protected]
```

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

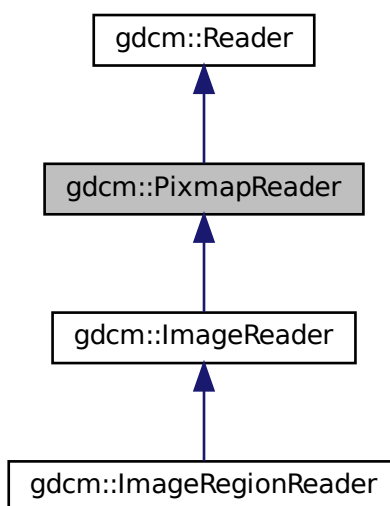
- [gdcMPixmap.h](#)

## 10.230 gdcm::PixmapReader Class Reference

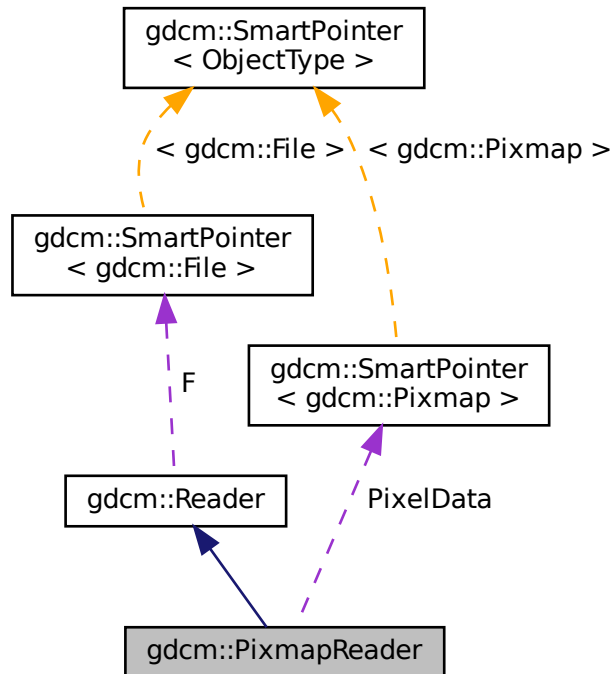
[PixmapReader.](#)

```
#include <gdcmPixmapReader.h>
```

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for `gdcm::PixmapReader`:



## Public Member Functions

- [PixmapReader](#) ()
- virtual [~PixmapReader](#) ()
- const [Pixmap](#) & [GetPixmap](#) () const  
*Return the read image (need to call [Read\(\)](#) first)*
- [Pixmap](#) & [GetPixmap](#) ()
- virtual bool [Read](#) ()

## Protected Member Functions

- virtual bool [ReadACRNEMAIImage](#) ()
- virtual bool [ReadImage](#) ([MediaStorage](#) const &ms)
- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

## Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

### 10.230.1 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](#)

### 10.230.2 Constructor & Destructor Documentation

#### 10.230.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ( )
```

#### 10.230.2.2 ~PixmapReader()

```
virtual gdcm::PixmapReader::~~PixmapReader ( ) [virtual]
```

### 10.230.3 Member Function Documentation

#### 10.230.3.1 GetPixmap() [1/2]

```
const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const
```

Return the read image (need to call [Read\(\)](#) first)

### 10.230.3.2 GetPixmap() [2/2]

```
Pixmap& gdcm::PixmapReader::GetPixmap ( )
```

### 10.230.3.3 Read()

```
virtual bool gdcm::PixmapReader::Read ( ) [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](#).

Reimplemented in [gdcm::ImageRegionReader](#), and [gdcm::ImageReader](#).

### 10.230.3.4 ReadACRNEMAIImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAIImage ( ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.230.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
    MediaStorage const & ms ) [protected], [virtual]
```

Reimplemented in [gdcm::ImageReader](#).

### 10.230.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
    MediaStorage const & ms,
    bool handlepixeldata = true ) [protected]
```



#### 10.230.4 Member Data Documentation

##### 10.230.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]
```

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

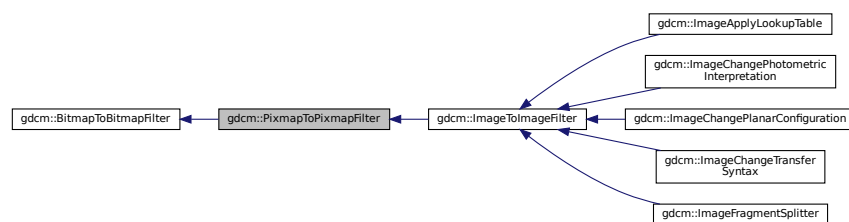
- [gdcmPixmapReader.h](#)

## 10.231 gdcm::PixmapToPixmapFilter Class Reference

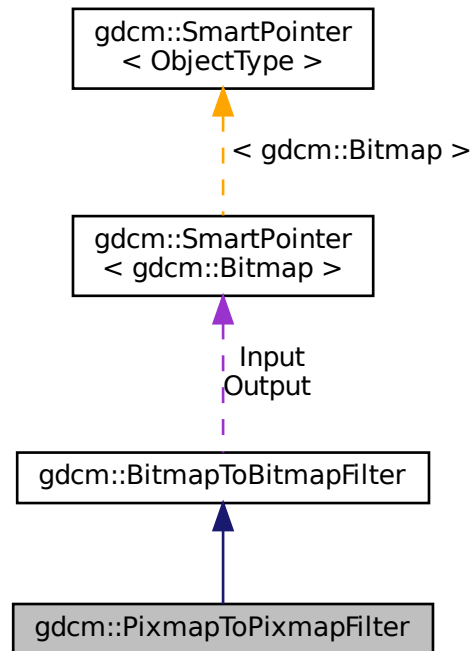
[PixmapToPixmapFilter](#) class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for `gdcm::PixmapToPixmapFilter`:



## Public Member Functions

- [PixmapToPixmapFilter \(\)](#)
- [~PixmapToPixmapFilter \(\)](#)
- [Pixmap & GetInput \(\)](#)
- [const Pixmap & GetOutput \(\) const](#)  
*Get Output image.*
- [const Pixmap & GetOutputAsPixmap \(\) const](#)

## Additional Inherited Members

### 10.231.1 Detailed Description

[PixmapToPixmapFilter](#) class.

Super class for all filter taking an image and producing an output image

## 10.231.2 Constructor & Destructor Documentation

### 10.231.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )
```

### 10.231.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ( ) [inline]
```

## 10.231.3 Member Function Documentation

### 10.231.3.1 GetInput()

```
Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )
```

### 10.231.3.2 GetOutput()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const
```

Get Output image.

### 10.231.3.3 GetOutputAsPixmap()

```
const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const
```

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

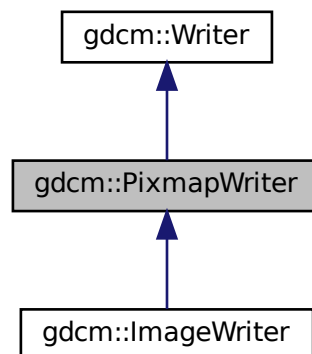
- [gdcmPixmapToPixmapFilter.h](#)

## 10.232 gdcm::PixmapWriter Class Reference

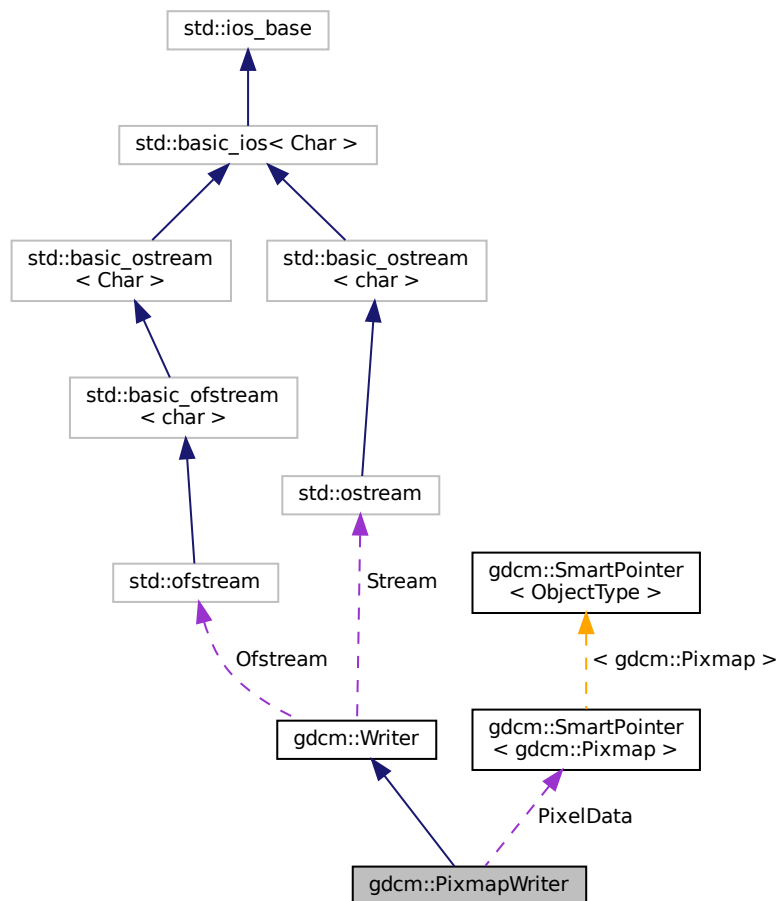
[PixmapWriter](#).

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



## Public Member Functions

- `PixmapWriter ()`
- `~PixmapWriter ()`
- `virtual const Pixmap & GetImage () const`
- `virtual Pixmap & GetImage ()`
- `const Pixmap & GetPixmap () const`
- `Pixmap & GetPixmap ()`
- `virtual void SetImage (Pixmap const &img)`
- `void SetPixmap (Pixmap const &img)`
- `bool Write ()`

*Write.*

## Protected Member Functions

- void [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ([MediaStorage](#) const &refms)

## Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

### 10.232.1 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)

### 10.232.2 Constructor & Destructor Documentation

#### 10.232.2.1 [PixmapWriter](#)()

```
gdcm::PixmapWriter::PixmapWriter ( )
```

#### 10.232.2.2 [~PixmapWriter](#)()

```
gdcm::PixmapWriter::~~PixmapWriter ( )
```

### 10.232.3 Member Function Documentation

### 10.232.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
    DataSet & ds,
    Pixmap const & image ) [protected]
```

### 10.232.3.2 GetImage() [1/2]

```
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](#).

### 10.232.3.3 GetImage() [2/2]

```
virtual Pixmap& gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]
```

Reimplemented in [gdcm::ImageWriter](#).

### 10.232.3.4 GetPixmap() [1/2]

```
const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const [inline]
```

### 10.232.3.5 GetPixmap() [2/2]

```
Pixmap& gdcm::PixmapWriter::GetPixmap ( ) [inline]
```

### 10.232.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (
    MediaStorage const & refs ) [protected]
```

### 10.232.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
    Pixmap const & img ) [virtual]
```

#### Examples:

[CompressImage.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#),  
and [TemplateEmptyImage.cxx](#).

### 10.232.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
    Pixmap const & img )
```

### 10.232.3.9 Write()

```
bool gdcm::PixmapWriter::Write ( ) [virtual]
```

Write.

Reimplemented from [gdcm::Writer](#).

## 10.232.4 Member Data Documentation

### 10.232.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]
```

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

- [gdcmPixmapWriter.h](#)

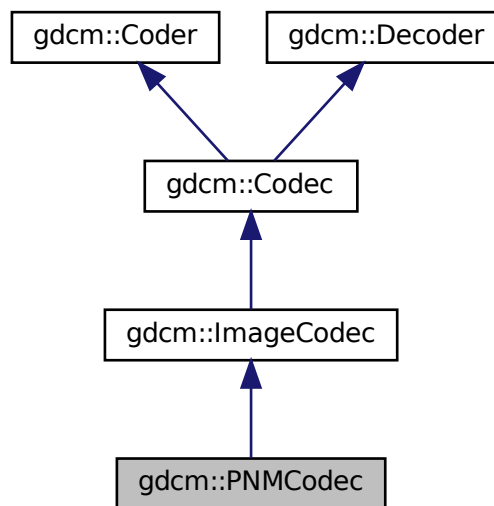


## 10.233 gdcm::PNMCodec Class Reference

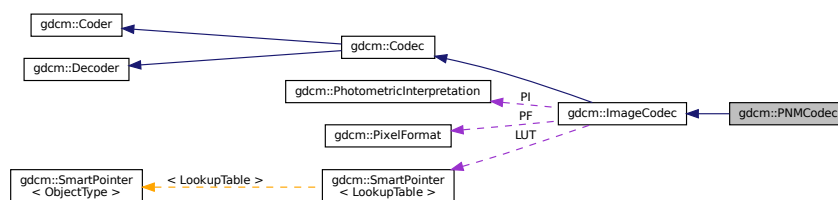
Class to do PNM.

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



### Public Member Functions

- [PNMCodec](#) ()
- [~PNMCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const

*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

*Return whether this decoder support this transfer syntax (can decode it)*

- virtual [ImageCodec](#) \* [Clone](#) () const
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- bool [Read](#) (const char \*filename, [DataElement](#) &out) const
- void [SetBufferLength](#) (unsigned long l)
- bool [Write](#) (const char \*filename, const [DataElement](#) &out) const

## Additional Inherited Members

### 10.233.1 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](#).

### 10.233.2 Constructor & Destructor Documentation

#### 10.233.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ( )
```

#### 10.233.2.2 ~PNMCodec()

```
gdcm::PNMCodec::~~PNMCodec ( )
```

### 10.233.3 Member Function Documentation

### 10.233.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.233.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.233.3.3 Clone()

```
virtual ImageCodec* gdcm::PNMCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.233.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]
```

### 10.233.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

#### 10.233.3.6 Read()

```
bool gdcM::PNMCodec::Read (
    const char * filename,
    DataElement & out ) const
```

#### 10.233.3.7 SetBufferLength()

```
void gdcM::PNMCodec::SetBufferLength (
    unsigned long l ) [inline]
```

#### 10.233.3.8 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](#)

## 10.234 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)

### 10.234.1 Detailed Description

DICOM [Preamble](#) (Part 10)

### 10.234.2 Constructor & Destructor Documentation

**10.234.2.1 Preamble()** [1/2]

```
gdcM::Preamble::Preamble ( )
```

**10.234.2.2 ~Preamble()**

```
gdcM::Preamble::~~Preamble ( )
```

**10.234.2.3 Preamble()** [2/2]

```
gdcM::Preamble::Preamble (
    Preamble const & ) [inline]
```

**10.234.3 Member Function Documentation****10.234.3.1 Clear()**

```
void gdcM::Preamble::Clear ( )
```

Clear.

**10.234.3.2 Create()**

```
void gdcM::Preamble::Create ( )
```

**10.234.3.3 GetInternal()**

```
const char* gdcM::Preamble::GetInternal ( ) const [inline]
```

Get internal pointer to preamble.

#### 10.234.3.4 GetLength()

```
VL gdcm::Preamble::GetLength ( ) const [inline]
```

Return size of [Preamble](#).

#### 10.234.3.5 IsEmpty()

```
bool gdcm::Preamble::IsEmpty ( ) const [inline]
```

Check if [Preamble](#) is empty.

#### 10.234.3.6 IsValid()

```
bool gdcm::Preamble::IsValid ( ) const [inline], [protected]
```

#### 10.234.3.7 operator=()

```
Preamble& gdcm::Preamble::operator= (
    Preamble const & ) [inline]
```

#### 10.234.3.8 Print()

```
void gdcm::Preamble::Print (
    std::ostream & os ) const
```

Print [Preamble](#).

#### 10.234.3.9 Read()

```
std::istream& gdcm::Preamble::Read (
    std::istream & is )
```

Read [Preamble](#).

#### 10.234.3.10 Remove()

```
void gdcM::Preamble::Remove ( )
```

#### 10.234.3.11 Valid()

```
void gdcM::Preamble::Valid ( )
```

Set [Preamble](#) to the default one.

#### 10.234.3.12 Write()

```
std::ostream const& gdcM::Preamble::Write (
    std::ostream & os ) const
```

Write [Preamble](#).

### 10.234.4 Friends And Related Function Documentation

#### 10.234.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Preamble & _val ) [friend]
```

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

- [gdcMPreamble.h](#)

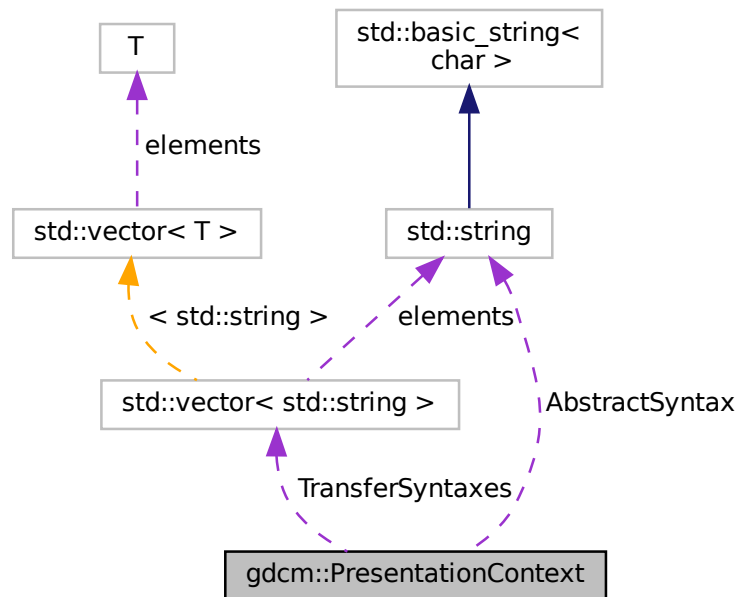


## 10.235 gdcm::PresentationContext Class Reference

[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::ImplicitVRLittleEndianDefault](#)↵  
[TransferSyntaxforDICOM](#))
- 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](#)

### 10.235.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

### 10.235.2 Member Typedef Documentation

#### 10.235.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

#### 10.235.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

### 10.235.3 Constructor & Destructor Documentation

#### 10.235.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ( )
```

### 10.235.3.2 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 ).

## 10.235.4 Member Function Documentation

### 10.235.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
    const char * tsstr )
```

### 10.235.4.2 GetAbstractSyntax()

```
const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]
```

### 10.235.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]
```

### 10.235.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const
```

### 10.235.4.5 GetTransferSyntax()

```
const char* gdcm::PresentationContext::GetTransferSyntax (
    SizeType i ) const [inline]
```

#### 10.235.4.6 operator==()

```
bool gdcmm::PresentationContext::operator== (
    const PresentationContext & pc ) const [inline]
```

References AbstractSyntax, and TransferSyntaxes.

#### 10.235.4.7 Print()

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

#### 10.235.4.8 SetAbstractSyntax()

```
void gdcmm::PresentationContext::SetAbstractSyntax (
    const char * absyn ) [inline]
```

#### 10.235.4.9 SetPresentationContextID()

```
void gdcmm::PresentationContext::SetPresentationContextID (
    uint8_t id )
```

### 10.235.5 Member Data Documentation

#### 10.235.5.1 AbstractSyntax

```
std::string gdcmm::PresentationContext::AbstractSyntax [protected]
```

Referenced by operator==().

#### 10.235.5.2 ID

```
uint8_t gdcmm::PresentationContext::ID [protected]
```

## 10.235.5.3 TransferSyntaxes

```
std::vector<std::string> gdcm::PresentationContext::TransferSyntaxes [protected]
```

Referenced by operator==( ).

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

- [gdcmPresentationContext.h](#)

## 10.236 gdcm::network::PresentationContextAC Class Reference

[PresentationContextAC](#).

```
#include <gdcmPresentationContextAC.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

## 10.236.1 Detailed Description

[PresentationContextAC](#).

[Table 9-18](#) PRESENTATION CONTEXT ITEM FIELDS

See also

[PresentationContext](#)

## 10.236.2 Constructor &amp; Destructor Documentation

### 10.236.2.1 PresentationContextAC()

```
gdcmm::network::PresentationContextAC::PresentationContextAC ( )
```

## 10.236.3 Member Function Documentation

### 10.236.3.1 GetPresentationContextID()

```
uint8_t gdcmm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]
```

### 10.236.3.2 GetReason()

```
uint8_t gdcmm::network::PresentationContextAC::GetReason ( ) const [inline]
```

### 10.236.3.3 GetTransferSyntax()

```
TransferSyntaxSub const& gdcmm::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]
```

### 10.236.3.4 Print()

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

### 10.236.3.5 Read()

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

#### 10.236.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (
    uint8_t id )
```

#### 10.236.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (
    uint8_t r ) [inline]
```

#### 10.236.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (
    TransferSyntaxSub const & ts )
```

#### 10.236.3.9 Size()

```
size_t gdcm::network::PresentationContextAC::Size ( ) const
```

#### 10.236.3.10 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](#)

## 10.237 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

### 10.237.1 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 [SetMergeModeToTransferSyntax](#) merge any new [TransferSyntax](#) to the already existing [PresentationContext](#) in order to re-use the same [AbstractSyntax](#).

See also

[PresentationContext](#)

Examples:

[CStoreQtProgress.cxx](#).



## 10.237.2 Member Typedef Documentation

### 10.237.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContext↵  
ArrayType
```

### 10.237.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

## 10.237.3 Constructor & Destructor Documentation

### 10.237.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ( )
```

## 10.237.4 Member Function Documentation

### 10.237.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (   
    const File & file )
```

Add a single [PresentationContext](#) from a single [File](#). Call multiple times when dealing with multiple files.

### 10.237.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (   
    const char * absyn,   
    const char * ts ) [protected]
```

#### 10.237.4.3 GenerateFromFileNames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFileNames (
    const Directory::FilenameType & files )
```

Generate the [PresentationContext](#) array from a File-Set. [File](#) specified needs to be valid DICOM files. Used for C-ST↔ORE operations

Examples:

[CStoreQtProgress.cxx](#).

#### 10.237.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
    UIDs::TSName asname )
```

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

#### 10.237.4.5 GetDefaultTransferSyntax()

```
const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]
```

#### 10.237.4.6 GetPresentationContexts()

```
PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts (
) [inline]
```

Examples:

[CStoreQtProgress.cxx](#).

#### 10.237.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
    const TransferSyntax & ts )
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

## 10.237.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )
```

## 10.237.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )
```

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

- [gdcmPresentationContextGenerator.h](#)

## 10.238 gdcm::network::PresentationContextRQ Class Reference

[PresentationContextRQ](#).

```
#include <gdcmPresentationContextRQ.h>
```

## Public Types

- typedef std::vector< [TransferSyntaxSub](#) >::size\_type [SizeType](#)

## Public Member Functions

- [PresentationContextRQ](#) ()
- [PresentationContextRQ](#) (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefault, TransferSyntaxforDICOM)
- [PresentationContextRQ](#) (const [PresentationContext](#) &pc)
- void [AddTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- [AbstractSyntax](#) const & [GetAbstractSyntax](#) () const
- [AbstractSyntax](#) & [GetAbstractSyntax](#) ()
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8\_t [GetPresentationContextID](#) () const
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) ([SizeType](#) i) const
- [TransferSyntaxSub](#) & [GetTransferSyntax](#) ([SizeType](#) i)
- 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

### 10.238.1 Detailed Description

[PresentationContextRQ](#).

[Table 9-13 PRESENTATION CONTEXT ITEM FIELDS](#)

See also

[PresentationContextAC](#)

### 10.238.2 Member Typedef Documentation

#### 10.238.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

### 10.238.3 Constructor & Destructor Documentation

#### 10.238.3.1 [PresentationContextRQ\(\)](#) [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ( )
```

#### 10.238.3.2 [PresentationContextRQ\(\)](#) [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    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 ).

#### 10.238.3.3 [PresentationContextRQ\(\)](#) [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
    const PresentationContext & pc )
```

## 10.238.4 Member Function Documentation

### 10.238.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
    TransferSyntaxSub const & ts )
```

### 10.238.4.2 GetAbstractSyntax() [1/2]

```
AbstractSyntax const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]
```

### 10.238.4.3 GetAbstractSyntax() [2/2]

```
AbstractSyntax& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]
```

### 10.238.4.4 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]
```

### 10.238.4.5 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const
```

### 10.238.4.6 GetTransferSyntax() [1/2]

```
TransferSyntaxSub const& gdcm::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) const [inline]
```

**10.238.4.7 GetTransferSyntax()** [2/2]

```
TransferSyntaxSub& gdcM::network::PresentationContextRQ::GetTransferSyntax (
    SizeType i ) [inline]
```

**10.238.4.8 GetTransferSyntaxes()**

```
std::vector<TransferSyntaxSub> const& gdcM::network::PresentationContextRQ::GetTransferSyntaxes (
) const [inline]
```

**10.238.4.9 operator==()**

```
bool gdcM::network::PresentationContextRQ::operator== (
    const PresentationContextRQ & pc ) const [inline]
```

**10.238.4.10 Print()**

```
void gdcM::network::PresentationContextRQ::Print (
    std::ostream & os ) const
```

**10.238.4.11 Read()**

```
std::istream& gdcM::network::PresentationContextRQ::Read (
    std::istream & is )
```

**10.238.4.12 SetAbstractSyntax()**

```
void gdcM::network::PresentationContextRQ::SetAbstractSyntax (
    AbstractSyntax const & absyn )
```

**10.238.4.13 SetPresentationContextID()**

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
    uint8_t id )
```

**10.238.4.14 Size()**

```
size_t gdcm::network::PresentationContextRQ::Size ( ) const
```

**10.238.4.15 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](#)

**10.239 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)

### 10.239.1 Detailed Description

[PresentationDataValue](#).

[Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

### 10.239.2 Constructor & Destructor Documentation

#### 10.239.2.1 PresentationDataValue()

```
gdcM::network::PresentationDataValue::PresentationDataValue ( )
```

### 10.239.3 Member Function Documentation

#### 10.239.3.1 ConcatenatePDVBlobs()

```
static DataSet gdcM::network::PresentationDataValue::ConcatenatePDVBlobs (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```

#### Warning

[DataSet](#) will be read as Implicit Little Endian TS

#### 10.239.3.2 ConcatenatePDVBlobsAsExplicit()

```
static DataSet gdcM::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
    const std::vector< PresentationDataValue > & inPDVs ) [static]
```



### 10.239.3.3 GetBlob()

```
const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const
```

### 10.239.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const
```

### 10.239.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const
```

### 10.239.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]
```

### 10.239.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]
```

### 10.239.3.8 Print()

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

### 10.239.3.9 Read()

```
std::istream& gdcm::network::PresentationDataValue::Read (
    std::istream & is )
```

**10.239.3.10 ReadInto()**

```
std::istream& gdcmm::network::PresentationDataValue::ReadInto (
    std::istream & is,
    std::ostream & os )
```

**10.239.3.11 SetBlob()**

```
void gdcmm::network::PresentationDataValue::SetBlob (
    const std::string & partialblob )
```

**10.239.3.12 SetCommand()**

```
void gdcmm::network::PresentationDataValue::SetCommand (
    bool inCommand )
```

**10.239.3.13 SetDataSet()**

```
void gdcmm::network::PresentationDataValue::SetDataSet (
    const DataSet & ds )
```

Set [DataSet](#). Write [DataSet](#) in implicit.

**Warning**

size of dataset should be below maxpdusize

**10.239.3.14 SetLastFragment()**

```
void gdcmm::network::PresentationDataValue::SetLastFragment (
    bool inLast )
```

### 10.239.3.15 SetMessageHeader()

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
    uint8_t messageheader ) [inline]
```

### 10.239.3.16 SetPresentationContextID()

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
    uint8_t id ) [inline]
```

### 10.239.3.17 Size()

```
size_t gdcm::network::PresentationDataValue::Size ( ) const
```

### 10.239.3.18 Write()

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

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

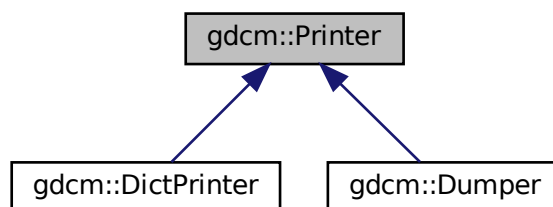
- [gdcmPresentationDataValue.h](#)

## 10.240 gdcm::Printer Class Reference

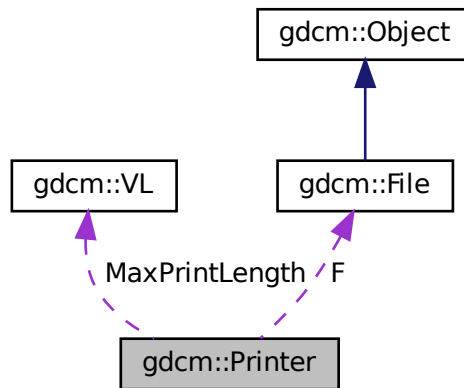
[Printer](#) class.

```
#include <gdcmPrinter.h>
```

Inheritance diagram for gdcm::Printer:



Collaboration diagram for `gdcm::Printer`:



## Public Types

- enum `PrintStyles` {  
`VERBOSE_STYLE` = 0,  
`CONDENSED_STYLE`,  
`XML`,  
`CXX` }

## Public Member Functions

- `Printer ()`
- `~Printer ()`
- `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

- [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](#)

### 10.240.1 Detailed Description

[Printer](#) class.

Examples:

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

### 10.240.2 Member Enumeration Documentation

#### 10.240.2.1 PrintStyles

enum [gdcm::Printer::PrintStyles](#)

Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

### 10.240.3 Constructor & Destructor Documentation

#### 10.240.3.1 Printer()

```
gdcm::Printer::Printer ( )
```

### 10.240.3.2 ~Printer()

```
gdcm::Printer::~~Printer ( )
```

## 10.240.4 Member Function Documentation

### 10.240.4.1 GetPrintStyle()

```
PrintStyle gdcm::Printer::GetPrintStyle ( ) const [inline]
```

Get PrintStyle value.

### 10.240.4.2 Print()

```
void gdcm::Printer::Print (
    std::ostream & os )
```

Print.

Examples:

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

### 10.240.4.3 PrintDataElement()

```
VR gdcm::Printer::PrintDataElement (
    std::ostringstream & os,
    const Dicts & dicts,
    const DataSet & ds,
    const DataElement & de,
    std::ostream & out,
    std::string const & indent ) [protected]
```

#### 10.240.4.4 PrintDataSet()

```
void gdcmm::Printer::PrintDataSet (
    const DataSet & ds,
    std::ostream & os,
    const std::string & s = "" )
```

Print an individual dataset.

#### 10.240.4.5 PrintSQ()

```
void gdcmm::Printer::PrintSQ (
    const SequenceOfItems * sqi,
    std::ostream & os,
    std::string const & indent ) [protected]
```

#### 10.240.4.6 SetColor()

```
void gdcmm::Printer::SetColor (
    bool c )
```

Set color mode or not.

Examples:

[DumpToshibaDTI.cxx](#).

#### 10.240.4.7 SetFile()

```
void gdcmm::Printer::SetFile (
    File const & f ) [inline]
```

Set file.

Examples:

[DumpSiemensBase64.cxx](#), and [DumpToshibaDTI.cxx](#).

#### 10.240.4.8 SetStyle()

```
void gdcM::Printer::SetStyle (
    PrintStyles ps ) [inline]
```

Set PrintStyle value.

### 10.240.5 Member Data Documentation

#### 10.240.5.1 F

```
const File* gdcM::Printer::F [protected]
```

#### 10.240.5.2 MaxPrintLength

```
VL gdcM::Printer::MaxPrintLength [protected]
```

#### 10.240.5.3 PrintStyle

```
PrintStyles gdcM::Printer::PrintStyle [protected]
```

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

- [gdcMPrinter.h](#)

## 10.241 gdcM::PrivateDict Class Reference

Private [Dict](#).

```
#include <gdcMDict.h>
```



## Public Member Functions

- [PrivateDict](#) ()
- [~PrivateDict](#) ()
- 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)

### 10.241.1 Detailed Description

Private [Dict](#).

### 10.241.2 Constructor & Destructor Documentation

#### 10.241.2.1 PrivateDict()

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

#### 10.241.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~~PrivateDict ( ) [inline]
```

### 10.241.3 Member Function Documentation

#### 10.241.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
    const PrivateTag & tag,
    const DictEntry & de ) [inline]
```

References `gdcm::DictEntry::GetVM()`, `gdcm::DictEntry::GetVR()`, `gdcm::DictEntry::SetVR()`, and `gdcm::VR::UN`.

#### 10.241.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
    const PrivateTag & tag ) const [inline]
```

#### 10.241.3.3 GetDictEntry()

```
const DictEntry& gdcm::PrivateDict::GetDictEntry (
    const PrivateTag & tag ) const [inline]
```

#### 10.241.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty ( ) const [inline]
```

#### 10.241.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ( ) [protected]
```

#### 10.241.3.6 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()`.

### 10.241.3.7 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.

## 10.241.4 Friends And Related Function Documentation

### 10.241.4.1 Dicts

```
friend class Dicts [friend]
```

### 10.241.4.2 operator<<

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

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

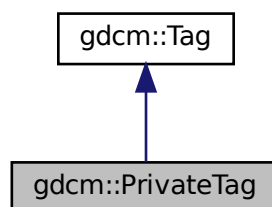
- [gdcmDict.h](#)

## 10.242 gdcm::PrivateTag Class Reference

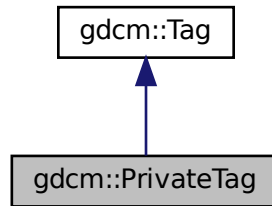
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for `gdcm::PrivateTag`:



### Public Member Functions

- [PrivateTag](#) (`uint16_t` group=0, `uint16_t` element=0, `const char *`owner="")
- [PrivateTag](#) ([Tag](#) `const &t`, `const char *`owner="")
- [DataElement](#) [GetAsDataElement](#) () `const`
- `const char *` [GetOwner](#) () `const`
- `bool` [operator<](#) (`const` [PrivateTag](#) `&_val`) `const`
- `bool` [ReadFromCommaSeparatedString](#) (`const char *`str)
- `void` [SetOwner](#) (`const char *`owner)

### Friends

- `std::ostream &` [operator<<](#) (`std::ostream &_os`, `const` [PrivateTag](#) `&_val`)

### 10.242.1 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](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [ELSC←INT1WaveToText.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [MrProtocol.cxx](#), [pmsct\\_rgb1.cxx](#), [Public←Dict.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

### 10.242.2 Constructor & Destructor Documentation

### 10.242.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (
    uint16_t group = 0,
    uint16_t element = 0,
    const char * owner = "" ) [inline]
```

### 10.242.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (
    Tag const & t,
    const char * owner = "" ) [inline]
```

References gdcm::Tag::GetElement().

## 10.242.3 Member Function Documentation

### 10.242.3.1 GetAsDataElement()

```
DataElement gdcm::PrivateTag::GetAsDataElement ( ) const
```

### 10.242.3.2 GetOwner()

```
const char* gdcm::PrivateTag::GetOwner ( ) const [inline]
```

Examples:

[PublicDict.cxx](#).

Referenced by gdcm::PrivateDict::PrintXML().

### 10.242.3.3 operator<()

```
bool gdcm::PrivateTag::operator< (
    const PrivateTag & _val ) const
```

#### 10.242.3.4 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")

#### 10.242.3.5 SetOwner()

```
void gdcM::PrivateTag::SetOwner (
    const char * owner ) [inline]
```

### 10.242.4 Friends And Related Function Documentation

#### 10.242.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const PrivateTag & _val ) [friend]
```

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

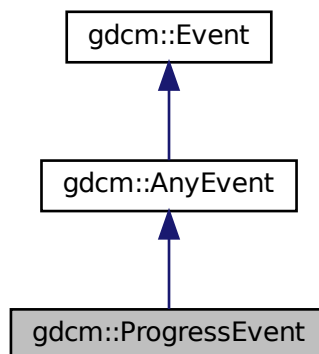
- [gdcMPrivateTag.h](#)

## 10.243 gdcM::ProgressEvent Class Reference

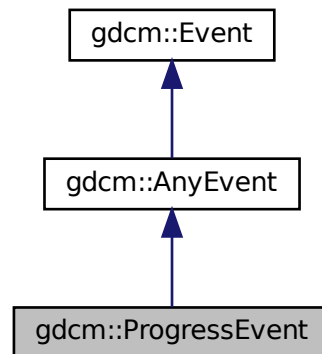
[ProgressEvent](#).

```
#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](#) (double p=0)
- [ProgressEvent](#) (const [Self](#) &s)
- virtual [~ProgressEvent](#) ()
- virtual bool [CheckEvent](#) (const [::gdcM::Event](#) \*e) const
- virtual const char \* [GetEventName](#) () const
- double [GetProgress](#) () const
- virtual [::gdcM::Event](#) \* [MakeObject](#) () const
- void [SetProgress](#) (double p)

### 10.243.1 Detailed Description

[ProgressEvent](#).

Special type of event triggered during

See also

[AnyEvent](#)

## 10.243.2 Member Typedef Documentation

### 10.243.2.1 Self

```
typedef ProgressEvent gdcm::ProgressEvent::Self
```

### 10.243.2.2 Superclass

```
typedef AnyEvent gdcm::ProgressEvent::Superclass
```

## 10.243.3 Constructor & Destructor Documentation

### 10.243.3.1 [ProgressEvent\(\)](#) [1/2]

```
gdcm::ProgressEvent::ProgressEvent (  
    double p = 0 ) [inline]
```

### 10.243.3.2 [~ProgressEvent\(\)](#)

```
virtual gdcm::ProgressEvent::~~ProgressEvent ( ) [inline], [virtual]
```

### 10.243.3.3 [ProgressEvent\(\)](#) [2/2]

```
gdcm::ProgressEvent::ProgressEvent (  
    const Self & s ) [inline]
```

## 10.243.4 Member Function Documentation



#### 10.243.4.1 CheckEvent()

```
virtual bool gdcm::ProgressEvent::CheckEvent (
    const ::gdcm::Event * e ) const [inline], [virtual]
```

#### 10.243.4.2 GetEventName()

```
virtual const char* gdcm::ProgressEvent::GetEventName ( ) const [inline], [virtual]
```

Return the StringName associated with the event.

Implements [gdcm::Event](#).

#### 10.243.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress ( ) const [inline]
```

#### 10.243.4.4 MakeObject()

```
virtual ::gdcm::Event* gdcm::ProgressEvent::MakeObject ( ) const [inline], [virtual]
```

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

Implements [gdcm::Event](#).

#### 10.243.4.5 SetProgress()

```
void gdcm::ProgressEvent::SetProgress (
    double p ) [inline]
```

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

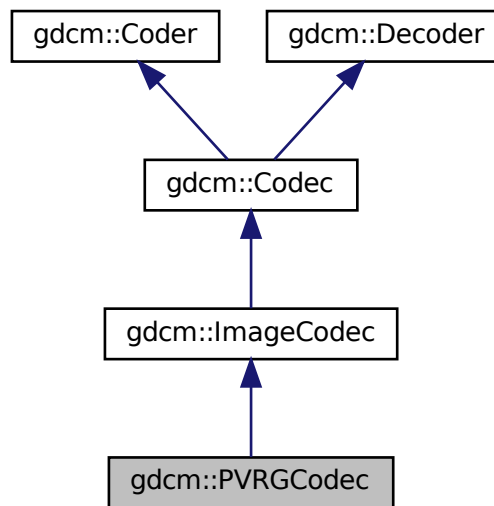
- [gdcmProgressEvent.h](#)

## 10.244 gdcm::PVRGCodec Class Reference

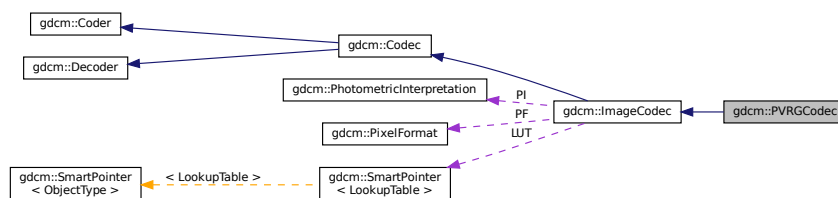
[PVRGCodec](#).

```
#include <gdcmPVRGCodec.h>
```

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



### Public Member Functions

- [PVRGCodec](#) ()
- [~PVRGCodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const

*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

*Return whether this decoder support this transfer syntax (can decode it)*

- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)

*Code.*

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

- void [SetLossyFlag](#) (bool l)

## Additional Inherited Members

### 10.244.1 Detailed Description

[PVRGCodec](#).

#### Note

pvrj 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\_Gyroscan-12-Jpeg\_Extended\_Process\_2\_4.dcm

we have to...

### 10.244.2 Constructor & Destructor Documentation

#### 10.244.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ( )
```

#### 10.244.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~~PVRGCodec ( )
```

### 10.244.3 Member Function Documentation

#### 10.244.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.244.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.244.3.3 Clone()

```
virtual ImageCodec* gdcm::PVRGCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.244.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

## 10.244.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

## 10.244.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
    bool l )
```

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

- [gdcmPVRGCodec.h](#)

## 10.245 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)

## 10.245.1 Detailed Description

[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.

## 10.245.2 Constructor & Destructor Documentation

### 10.245.2.1 PythonFilter()

```
gdcM::PythonFilter::PythonFilter ( )
```

### 10.245.2.2 ~PythonFilter()

```
gdcM::PythonFilter::~~PythonFilter ( )
```

## 10.245.3 Member Function Documentation

### 10.245.3.1 GetFile() [1/2]

```
File& gdcM::PythonFilter::GetFile ( ) [inline]
```

### 10.245.3.2 GetFile() [2/2]

```
const File& gdcM::PythonFilter::GetFile ( ) const [inline]
```

### 10.245.3.3 SetDicts()

```
void gdcM::PythonFilter::SetDicts (
    const Dicts & dicts )
```

### 10.245.3.4 SetFile()

```
void gdcM::PythonFilter::SetFile (
    const File & f ) [inline]
```

## 10.245.3.5 ToPyObject()

```
PyObject* gdcm::PythonFilter::ToPyObject (
    const Tag & t ) const
```

## 10.245.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
    bool ) [inline]
```

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

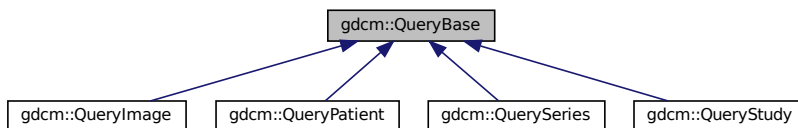
- [gdcmPythonFilter.h](#)

## 10.246 gdcm::QueryBase Class Reference

[QueryBase](#).

```
#include <gdcmQueryBase.h>
```

Inheritance diagram for gdcm::QueryBase:



## Public Member Functions

- virtual [~QueryBase](#) ()
- 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

### 10.246.1 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.

### 10.246.2 Constructor & Destructor Documentation

#### 10.246.2.1 ~QueryBase()

```
virtual gdcm::QueryBase::~~QueryBase ( ) [inline], [virtual]
```

### 10.246.3 Member Function Documentation

#### 10.246.3.1 GetAllRequiredTags()

```
std::vector<Tag> gdcm::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



### 10.246.3.2 GetAllTags()

```
std::vector<Tag> gdcm::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

### 10.246.3.3 GetHierachicalSearchTags()

```
virtual std::vector<Tag> gdcm::QueryBase::GetHierachicalSearchTags (
    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](#).

### 10.246.3.4 GetName()

```
virtual const char* gdcm::QueryBase::GetName ( ) const [pure virtual]
```

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

### 10.246.3.5 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](#).

### 10.246.3.6 GetQueryLevel()

```
virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const [pure virtual]
```

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

### 10.246.3.7 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](#).

### 10.246.3.8 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](#)

## 10.247 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](#) > &inCharSetType)
- static [BaseQuery](#) \* [ProduceQuery](#) (const std::string &sopInstanceUID, [ENQueryType](#) inQueryType)
- static [BaseRootQuery](#) \* [ProduceQuery](#) ([ERootType](#) inRootType, [EQueryType](#) inQueryType, [EQueryLevel](#) inQueryLevel)

### 10.247.1 Detailed Description

QueryFactory.h.

#### Note

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

## 10.247.2 Member Function Documentation

### 10.247.2.1 GetCharacterFromCurrentLocale()

```
static 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() ).

### 10.247.2.2 ListCharSets()

```
static void gdcm::QueryFactory::ListCharSets (
    std::ostream & os ) [static]
```

List all possible CharSet.

### 10.247.2.3 ProduceCharacterSetDataElement()

```
static 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

### 10.247.2.4 ProduceQuery() [1/2]

```
static BaseQuery* gdcm::QueryFactory::ProduceQuery (
    const std::string & sopInstanceUID,
    ENQueryType inQueryType ) [static]
```

### 10.247.2.5 ProduceQuery() [2/2]

```
static 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:

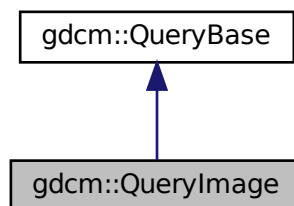
- [gdcMQueryFactory.h](#)

## 10.248 gdcM::QueryImage Class Reference

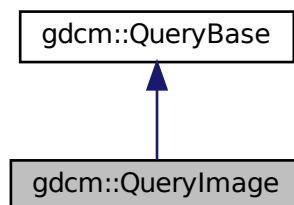
[QueryImage](#).

```
#include <gdcMQueryImage.h>
```

Inheritance diagram for gdcM::QueryImage:



Collaboration diagram for gdcM::QueryImage:



## Public Member Functions

- `std::vector< Tag > GetHierarchicalSearchTags` (const [ERootType](#) &inRootType) const  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const [ERootType](#) &inRootType) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const [ERootType](#) &inRootType) const
- `std::vector< Tag > GetUniqueTags` (const [ERootType](#) &inRootType) const

### 10.248.1 Detailed Description

[QueryImage](#).

contains: class to construct an image-based query for C-FIND and C-MOVE

### 10.248.2 Member Function Documentation

#### 10.248.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdcm::QueryImage::GetHierarchicalSearchTags (  
    const ERootType & inRootType ) const [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

#### 10.248.2.2 GetName()

```
const char* gdcm::QueryImage::GetName ( ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.248.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryImage::GetOptionalTags (  
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.248.2.4 GetQueryLevel()

```
DataElement gdcM::QueryImage::GetQueryLevel ( ) const [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.248.2.5 GetRequiredTags()

```
std::vector<Tag> gdcM::QueryImage::GetRequiredTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.248.2.6 GetUniqueTags()

```
std::vector<Tag> gdcM::QueryImage::GetUniqueTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcM::QueryBase](#).

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

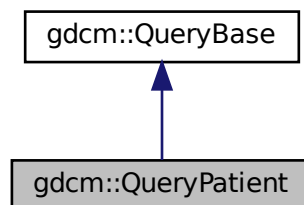
- [gdcMQueryImage.h](#)

## 10.249 gdcM::QueryPatient Class Reference

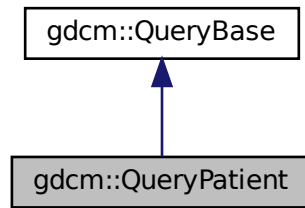
[QueryPatient](#).

```
#include <gdcMQueryPatient.h>
```

Inheritance diagram for [gdcM::QueryPatient](#):



Collaboration diagram for gdcm::QueryPatient:



## Public Member Functions

- `std::vector< Tag > GetHierarchicalSearchTags` (const `ERootType` &`inRootType`) const  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const

### 10.249.1 Detailed Description

[QueryPatient](#).

contains: class to construct a patient-based query for c-find and c-move

### 10.249.2 Member Function Documentation

#### 10.249.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdcm::QueryPatient::GetHierarchicalSearchTags (
    const ERootType & inRootType ) const [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

#### 10.249.2.2 GetName()

```
const char* gdcm::QueryPatient::GetName ( ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

#### 10.249.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QueryPatient::GetUniqueTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

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

- [gdcmQueryPatient.h](#)

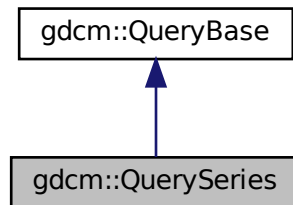


## 10.250 gdcm::QuerySeries Class Reference

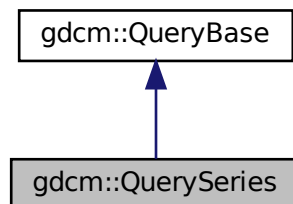
[QuerySeries.](#)

```
#include <gdcmQuerySeries.h>
```

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



### Public Member Functions

- `std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const`  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName () const`
- `std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const`
- `DataElement GetQueryLevel () const`
- `std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const`
- `std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const`

### 10.250.1 Detailed Description

[QuerySeries](#).

contains: class to construct a series-based query for c-find and c-move

### 10.250.2 Member Function Documentation

#### 10.250.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdcM::QuerySeries::GetHierarchicalSearchTags (
    const ERootType & inRootType ) const [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcM::QueryBase](#).

#### 10.250.2.2 GetName()

```
const char* gdcM::QuerySeries::GetName ( ) const [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.250.2.3 GetOptionalTags()

```
std::vector<Tag> gdcM::QuerySeries::GetOptionalTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcM::QueryBase](#).

#### 10.250.2.4 GetQueryLevel()

```
DataElement gdcM::QuerySeries::GetQueryLevel ( ) const [virtual]
```

Implements [gdcM::QueryBase](#).

### 10.250.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.250.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QuerySeries::GetUniqueTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

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

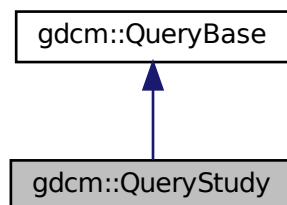
- [gdcmQuerySeries.h](#)

## 10.251 gdcm::QueryStudy Class Reference

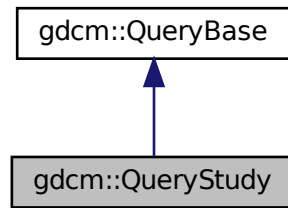
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for `gdcm::QueryStudy`:



Collaboration diagram for `gdcm::QueryStudy`:



## Public Member Functions

- `std::vector< Tag > GetHierarchicalSearchTags` (const `ERootType` &`inRootType`) const  
*Return all Unique Key for a particular Query Root type (from the same level and above).*
- `const char * GetName` () const
- `std::vector< Tag > GetOptionalTags` (const `ERootType` &`inRootType`) const
- `DataElement GetQueryLevel` () const
- `std::vector< Tag > GetRequiredTags` (const `ERootType` &`inRootType`) const
- `std::vector< Tag > GetUniqueTags` (const `ERootType` &`inRootType`) const

### 10.251.1 Detailed Description

`QueryStudy.h`.

contains: class to construct a study-based query for C-FIND and C-MOVE

### 10.251.2 Member Function Documentation

#### 10.251.2.1 GetHierarchicalSearchTags()

```
std::vector<Tag> gdcm::QueryStudy::GetHierarchicalSearchTags (
    const ERootType & inRootType ) const [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements `gdcm::QueryBase`.

### 10.251.2.2 GetName()

```
const char* gdcm::QueryStudy::GetName ( ) const [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.251.2.3 GetOptionalTags()

```
std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.251.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.251.2.5 GetRequiredTags()

```
std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

### 10.251.2.6 GetUniqueTags()

```
std::vector<Tag> gdcm::QueryStudy::GetUniqueTags (
    const ERootType & inRootType ) const [virtual]
```

Implements [gdcm::QueryBase](#).

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

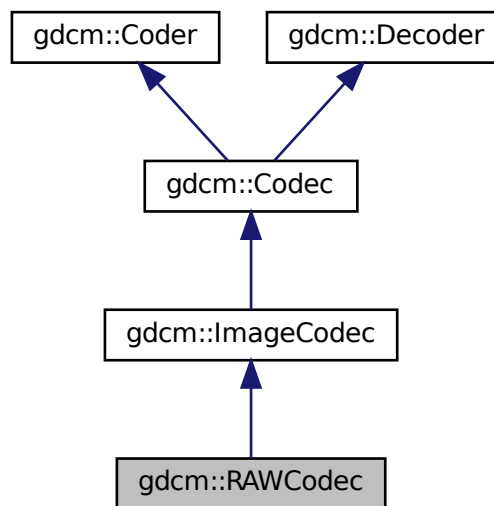
- [gdcmQueryStudy.h](#)

## 10.252 gdcm::RAWCodec Class Reference

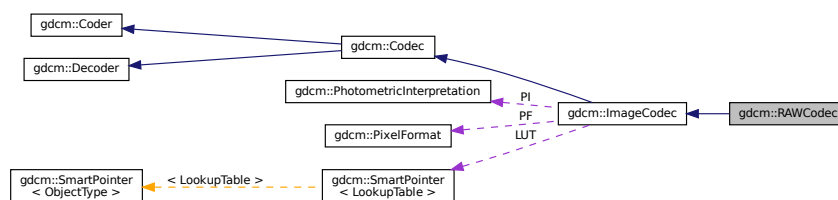
[RAWCodec](#) class.

```
#include <gdcmRAWCodec.h>
```

Inheritance diagram for `gdcm::RAWCodec`:



Collaboration diagram for `gdcm::RAWCodec`:



### Public Member Functions

- [RAWCodec](#) ()
- [~RAWCodec](#) ()
- `bool` [CanCode](#) ([TransferSyntax](#) const &ts) const

*Return whether this coder support this transfer syntax (can code it)*

- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const

*Return whether this decoder support this transfer syntax (can decode it)*

- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)

*Code.*

- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

- bool [DecodeBytes](#) (const char \*inBytes, size\_t inBufferLength, char \*outBytes, size\_t inOutBufferLength)
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)

## Protected Member Functions

- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)

## Additional Inherited Members

### 10.252.1 Detailed Description

[RAWCodec](#) class.

### 10.252.2 Constructor & Destructor Documentation

#### 10.252.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ( )
```

#### 10.252.2.2 ~RAWCodec()

```
gdcm::RAWCodec::~~RAWCodec ( )
```

### 10.252.3 Member Function Documentation

#### 10.252.3.1 CanCode()

```
bool gdcm::RAWCodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.252.3.2 CanDecode()

```
bool gdcm::RAWCodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

#### 10.252.3.3 Clone()

```
virtual ImageCodec* gdcm::RAWCodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

#### 10.252.3.4 Code()

```
bool gdcm::RAWCodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).



### 10.252.3.5 Decode()

```
bool gdcm::RAWCodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.252.3.6 DecodeByStreams()

```
bool gdcm::RAWCodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.252.3.7 DecodeBytes()

```
bool gdcm::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.

### 10.252.3.8 GetHeaderInfo()

```
bool gdcm::RAWCodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

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

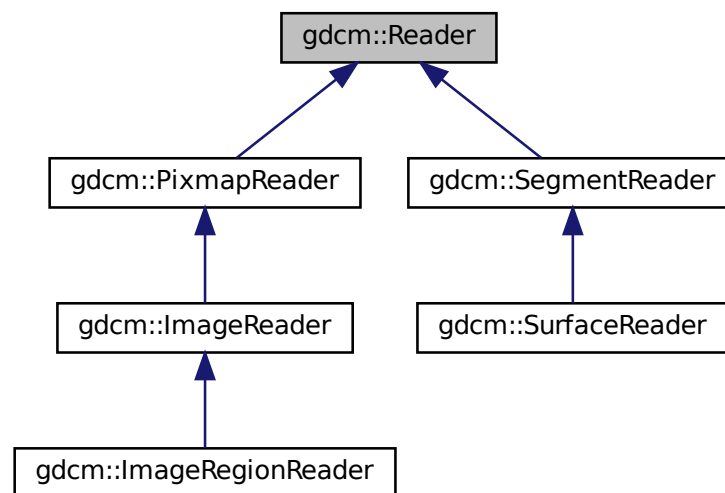
- [gdcmRAWCodec.h](#)

## 10.253 gdcm::Reader Class Reference

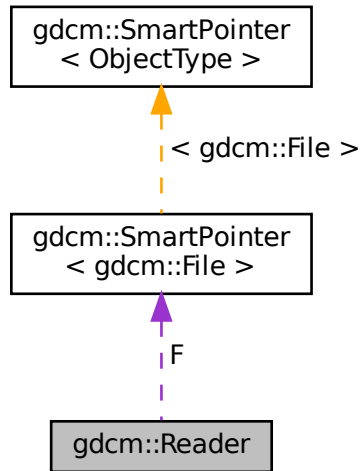
[Reader](#) ala DOM (Document [Object](#) Model)

```
#include <gdcmReader.h>
```

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



## Public Member Functions

- [Reader](#) ()
- virtual [~Reader](#) ()
- bool [CanRead](#) () const
- const [File](#) & [GetFile](#) () const  
*Set/Get File.*
- [File](#) & [GetFile](#) ()  
*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](#)

## 10.253.1 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](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [gdcmrtnplan.cxx](#), [gdcmrtpplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSEExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

## 10.253.2 Constructor & Destructor Documentation

### 10.253.2.1 Reader()

```
gdcm::Reader::Reader ( )
```

### 10.253.2.2 ~Reader()

```
virtual gdcm::Reader::~Reader ( ) [virtual]
```

## 10.253.3 Member Function Documentation

### 10.253.3.1 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](#).

### 10.253.3.2 GetFile() [1/2]

```
const File& gdcm::Reader::GetFile ( ) const [inline]
```

Set/Get [File](#).

#### Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

**10.253.3.3 GetFile()** [2/2]

```
File& gdcm::Reader::GetFile ( ) [inline]
```

Set/Get [File](#).

**10.253.3.4 GetStreamCurrentPosition()**

```
size_t gdcm::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++

**10.253.3.5 GetStreamPtr()**

```
std::istream* gdcm::Reader::GetStreamPtr ( ) const [inline], [protected]
```

**10.253.3.6 Read()**

```
virtual bool gdcm::Reader::Read ( ) [virtual]
```

Main function to read a file.

Reimplemented in [gdcm::ImageRegionReader](#), [gdcm::PixmapReader](#), [gdcm::ImageReader](#), [gdcm::SegmentReader](#), and [gdcm::SurfaceReader](#).

**Examples:**

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

### 10.253.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ( ) [protected]
```

### 10.253.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ( ) [protected]
```

### 10.253.3.9 ReadPreamble()

```
bool gdcm::Reader::ReadPreamble ( ) [protected]
```

### 10.253.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (
    std::set< PrivateTag > const & ptags,
    bool readvalues = true )
```

Will only read the specified selected private tags.

### 10.253.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
    std::set< Tag > const & tags,
    bool readvalues = true )
```

Will only read the specified selected tags.

### 10.253.3.12 ReadUpToTag()

```
bool gdcm::Reader::ReadUpToTag (
    const Tag & tag,
    std::set< Tag > const & skiptags = std::set< Tag >() )
```

Will read only up to Tag

## Parameters

<i>tag</i>	and skipping any tag specified in
<i>skiptags</i>	

## 10.253.3.13 SetFile()

```
void gdcm::Reader::SetFile (
    File & file ) [inline]
```

Set/Get [File](#).

## 10.253.3.14 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:

[BasicImageAnonymizer.cs](#), [ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpCSA.cs](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpSiemensBase64.cxx](#), [DumpToshibaDTI.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [SimplePrintPatientName.cs](#), [TemplateEmptyImage.cxx](#), [TestReader.cxx](#), and [threadgdcm.cxx](#).

## 10.253.3.15 SetStream()

```
void gdcm::Reader::SetStream (
    std::istream & input_stream ) [inline]
```

Set the open-ed stream directly.

## Examples:

[DumpToshibaDTI.cxx](#), and [ReadUTF8QtDir.cxx](#).



## 10.253.4 Friends And Related Function Documentation

### 10.253.4.1 StreamImageReader

```
friend class StreamImageReader [friend]
```

## 10.253.5 Member Data Documentation

### 10.253.5.1 F

```
SmartPointer<File> gdcm::Reader::F [protected]
```

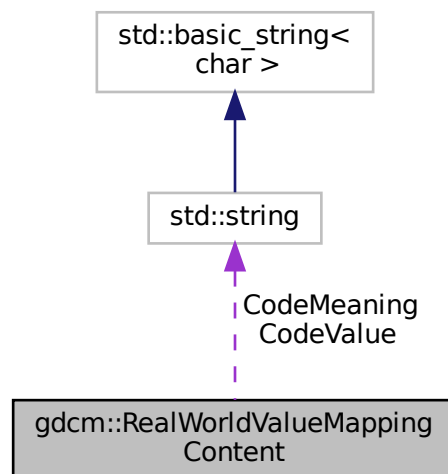
The documentation for this class was generated from the following file:

- [gdcmReader.h](#)

## 10.254 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for gdcm::RealWorldValueMappingContent:



## Public Attributes

- `std::string` [CodeMeaning](#)
- `std::string` [CodeValue](#)
- `double` [RealWorldValueIntercept](#)
- `double` [RealWorldValueSlope](#)

## 10.254.1 Member Data Documentation

### 10.254.1.1 CodeMeaning

```
std::string gdcmm::RealWorldValueMappingContent::CodeMeaning
```

### 10.254.1.2 CodeValue

```
std::string gdcmm::RealWorldValueMappingContent::CodeValue
```

### 10.254.1.3 RealWorldValueIntercept

```
double gdcmm::RealWorldValueMappingContent::RealWorldValueIntercept
```

### 10.254.1.4 RealWorldValueSlope

```
double gdcmm::RealWorldValueMappingContent::RealWorldValueSlope
```

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

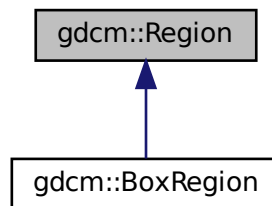
- [gdcmmImageHelper.h](#)

## 10.255 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.*

### 10.255.1 Detailed Description

Class for manipulation region.

### 10.255.2 Constructor & Destructor Documentation

### 10.255.2.1 Region()

```
gdcM::Region::Region ( )
```

### 10.255.2.2 ~Region()

```
virtual gdcM::Region::~~Region ( ) [virtual]
```

## 10.255.3 Member Function Documentation

### 10.255.3.1 Area()

```
virtual size_t gdcM::Region::Area ( ) const [pure virtual]
```

compute the area

Implemented in [gdcM::BoxRegion](#).

### 10.255.3.2 Clone()

```
virtual Region\* gdcM::Region::Clone ( ) const [pure virtual]
```

Implemented in [gdcM::BoxRegion](#).

### 10.255.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcM::Region::ComputeBoundingBox ( ) [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcM::BoxRegion](#).

#### 10.255.3.4 Empty()

```
virtual bool gdcm::Region::Empty ( ) const [pure virtual]
```

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

#### 10.255.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid ( ) const [pure virtual]
```

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

#### 10.255.3.6 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](#)

## 10.256 gdcm::Rescaler Class Reference

Rescale class.

```
#include <gdcmRescaler.h>
```

## Public Member Functions

- [Rescaler](#) ()
- [~Rescaler](#) ()
- [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)

### 10.256.1 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 ouput, 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](#)

## 10.256.2 Constructor & Destructor Documentation

### 10.256.2.1 Rescaler()

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

### 10.256.2.2 ~Rescaler()

```
gdcm::Rescaler::~~Rescaler ( ) [inline]
```

## 10.256.3 Member Function Documentation

### 10.256.3.1 ComputeInterceptSlopePixelFormat()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelFormat ( )
```

Compute the Pixel Format of the output data Used for direct transformation

### 10.256.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )
```

Compute the Pixel Format of the output data Used for inverse transformation

### 10.256.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept ( ) const [inline]
```

### 10.256.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope ( ) const [inline]
```

### 10.256.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
    char * out,
    const char * in,
    size_t n )
```

Inverse transform.

### 10.256.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.256.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
    char * out,
    const char * in,
    size_t n )
```

Direct transform.



### 10.256.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn >
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
    char * out,
    const TIn * in,
    size_t n ) [protected]
```

### 10.256.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
    double i ) [inline]
```

Set Intercept: used for both direct&inverse transformation.

### 10.256.3.10 SetMinMaxForPixelType()

```
void gdcm::Rescaler::SetMinMaxForPixelType (
    double min,
    double max ) [inline]
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

### 10.256.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
    PixelFormat const & pf ) [inline]
```

Set Pixel Format of input data.

### 10.256.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
    double s ) [inline]
```

Set Slope: user for both direct&inverse transformation.

### 10.256.3.13 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 specifix Target Pixel [Type](#)

### 10.256.3.14 SetUseTargetPixelFormat()

```
void gdcM::Rescaler::SetUseTargetPixelFormat (
    bool b )
```

Override default behavior of Rescale.

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

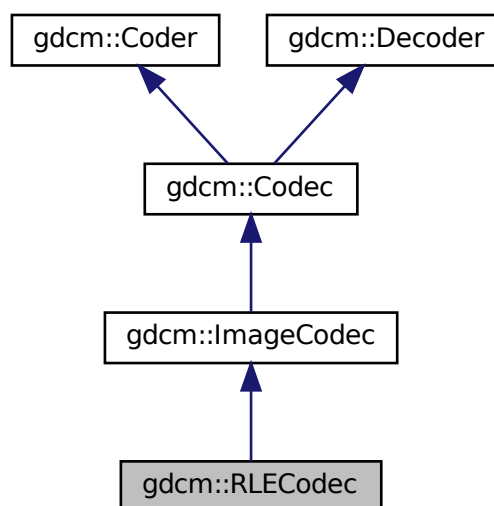
- [gdcMRescaler.h](#)

## 10.257 gdcM::RLECodec Class Reference

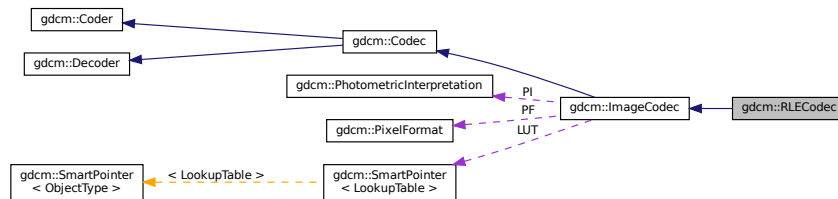
Class to do RLE.

```
#include <gdcMRLECodec.h>
```

Inheritance diagram for gdcM::RLECodec:



Collaboration diagram for gdcm::RLECodec:



## Public Member Functions

- [RLECodec](#) ()
- [~RLECodec](#) ()
- bool [CanCode](#) ([TransferSyntax](#) const &ts) const  
*Return whether this coder support this transfer syntax (can code it)*
- bool [CanDecode](#) ([TransferSyntax](#) const &ts) const  
*Return whether this decoder support this transfer syntax (can decode it)*
- virtual [ImageCodec](#) \* [Clone](#) () const
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)  
*Code.*
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)  
*Decode.*
- unsigned long [GetBufferLength](#) () const
- bool [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- void [SetBufferLength](#) (unsigned long l)
- void [SetLength](#) (unsigned long l)

## Protected Member Functions

- bool [AppendFrameEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [AppendRowEncode](#) (std::ostream &out, const char \*data, size\_t datalen)
- bool [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- 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](#) ()
- bool [IsRowEncoder](#) ()
- bool [StartEncode](#) (std::ostream &)
- bool [StopEncode](#) (std::ostream &)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 10.257.1 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).

### 10.257.2 Constructor & Destructor Documentation

#### 10.257.2.1 RLECodec()

```
gdcm::RLECodec::RLECodec ( )
```

#### 10.257.2.2 ~RLECodec()

```
gdcm::RLECodec::~~RLECodec ( )
```

### 10.257.3 Member Function Documentation

#### 10.257.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
    std::ostream & out,
    const char * data,
    size_t datalen ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.3 CanCode()

```
bool gdcm::RLECodec::CanCode (
    TransferSyntax const & ) const [virtual]
```

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.4 CanDecode()

```
bool gdcm::RLECodec::CanDecode (
    TransferSyntax const & ) const [virtual]
```

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.5 Clone()

```
virtual ImageCodec* gdcm::RLECodec::Clone ( ) const [virtual]
```

Implements [gdcm::ImageCodec](#).

### 10.257.3.6 Code()

```
bool gdcm::RLECodec::Code (
    DataElement const & in_,
    DataElement & out_ ) [virtual]
```

Code.

Reimplemented from [gdcm::Coder](#).

### 10.257.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
    DataElement const & ,
    DataElement & ) [virtual]
```

Decode.

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
    std::istream & is,
    std::ostream & os ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.9 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]
```

### 10.257.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]
```

### 10.257.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
    std::istream & is,
    TransferSyntax & ts ) [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ( ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
    unsigned long l ) [inline]
```

### 10.257.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
    unsigned long l ) [inline]
```

### 10.257.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

### 10.257.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
    std::ostream & ) [protected], [virtual]
```

Reimplemented from [gdcm::ImageCodec](#).

## 10.257.4 Friends And Related Function Documentation

### 10.257.4.1 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

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

- [gdcmRLECodec.h](#)

## 10.258 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

### 10.258.1 Detailed Description

[RoleSelectionSub](#).

PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 10.258.2 Constructor & Destructor Documentation

#### 10.258.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ( )
```



### 10.258.3 Member Function Documentation

#### 10.258.3.1 Print()

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

#### 10.258.3.2 Read()

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

#### 10.258.3.3 SetTuple()

```
void gdcmm::network::RoleSelectionSub::SetTuple (
    const char * uid,
    uint8_t scurole,
    uint8_t scprole )
```

#### 10.258.3.4 Size()

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

#### 10.258.3.5 Write()

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

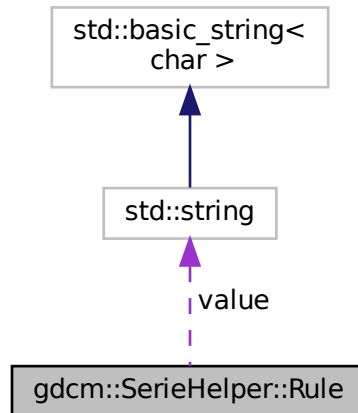
The documentation for this class was generated from the following file:

- [gdcmmRoleSelectionSub.h](#)

## 10.259 gdcm::SerieHelper::Rule Struct Reference

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper::Rule:



### Public Attributes

- `uint16_t` [elem](#)
- `uint16_t` [group](#)
- `int` [op](#)
- `std::string` [value](#)

### 10.259.1 Member Data Documentation

#### 10.259.1.1 elem

```
uint16_t gdcm::SerieHelper::Rule::elem
```

### 10.259.1.2 group

```
uint16_t gdcM::SerieHelper::Rule::group
```

### 10.259.1.3 op

```
int gdcM::SerieHelper::Rule::op
```

### 10.259.1.4 value

```
std::string gdcM::SerieHelper::Rule::value
```

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

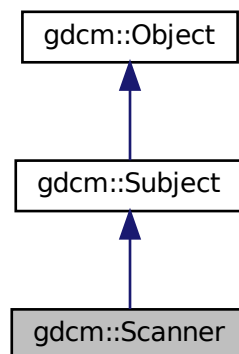
- [gdcMSerieHelper.h](#)

## 10.260 gdcM::Scanner Class Reference

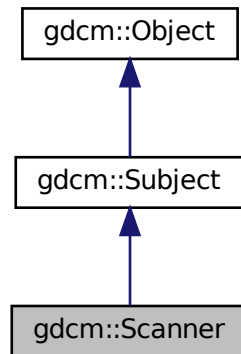
[Scanner.](#)

```
#include <gdcMScanner.h>
```

Inheritance diagram for gdcM::Scanner:



Collaboration diagram for `gdcm::Scanner`:



## Classes

- struct [lststr](#)

## Public Types

- typedef `MappingType::const_iterator` [ConstIterator](#)
- typedef `std::map< const char *, TagToValue, lststr >` [MappingType](#)
- typedef `std::map< Tag, const char * >` [TagToValue](#)
- typedef `TagToValue::value_type` [TagToValueValueType](#)
- typedef `std::set< std::string >` [ValuesType](#)

## Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) ()
- 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::FilenameType](#) [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  
*Print result.*
- bool [Scan](#) ([Directory::FilenamesType](#) const &filenames)  
*Start the scan !*

## 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)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Scanner](#) &s)

### 10.260.1 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](#).

**10.260.2 Member Typedef Documentation****10.260.2.1 ConstIterator**

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

**10.260.2.2 MappingType**

```
typedef std::map<const char *, TagToValue, ltstr> gdcm::Scanner::MappingType
```

**10.260.2.3 TagToValue**

```
typedef std::map<Tag, const char*> gdcm::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)`

#### 10.260.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

#### 10.260.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

### 10.260.3 Constructor & Destructor Documentation

#### 10.260.3.1 Scanner()

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

#### 10.260.3.2 ~Scanner()

```
gdcm::Scanner::~~Scanner ( )
```

### 10.260.4 Member Function Documentation

#### 10.260.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.260.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.260.4.3 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](#).

#### 10.260.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin ( ) const [inline]
```

#### 10.260.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ( )
```

#### 10.260.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ( )
```

#### 10.260.4.7 End()

```
ConstIterator gdcm::Scanner::End ( ) const [inline]
```

#### 10.260.4.8 GetAllFileNamesFromTagToValue()

```
Directory::FileNamesType gdcm::Scanner::GetAllFileNamesFromTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'



#### 10.260.4.9 GetFilenameFromTagToValue()

```
const char* gdcm::Scanner::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'

#### 10.260.4.10 GetFileNames()

```
Directory::FileNamesType const& gdcm::Scanner::GetFileNames ( ) const [inline]
```

#### 10.260.4.11 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](#).

#### 10.260.4.12 GetMapping()

```
TagToValue const& gdcm::Scanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples:

[DumpToSQLITE3.cxx](#).

#### 10.260.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.260.4.14 GetMappings()

```
MappingType const& gdcm::Scanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.260.4.15 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)

#### 10.260.4.16 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 !

#### 10.260.4.17 GetValues() [1/2]

```
ValueType const& gdcm::Scanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

#### Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.260.4.18 GetValues() [2/2]

```
ValueType gdcm::Scanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

#### 10.260.4.19 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](#).

#### 10.260.4.20 New()

```
static SmartPointer<Scanner> gdcm::Scanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

#### 10.260.4.21 Print()

```
void gdcm::Scanner::Print (
    std::ostream & os ) const [virtual]
```

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by [gdcm::operator<<\(\)](#).

#### 10.260.4.22 ProcessPublicTag()

```
void gdcM::Scanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.260.4.23 Scan()

```
bool gdcM::Scanner::Scan (
    Directory::FileNamesType const & filenames )
```

Start the scan !

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 10.260.5 Friends And Related Function Documentation

#### 10.260.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Scanner & s ) [friend]
```

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

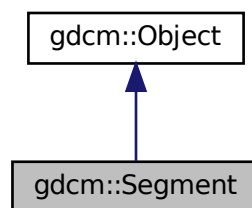
- [gdcMScanner.h](#)

## 10.261 gdcm::Segment Class Reference

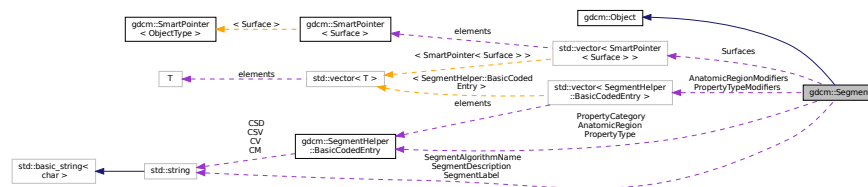
This class defines a segment.

```
#include <gdcmSegment.h>
```

Inheritance diagram for gdcm::Segment:



Collaboration diagram for `gdcm::Segment`:



## 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](#) ()
- virtual [~Segment](#) ()
- void [AddSurface](#) ([SmartPointer](#)< [Surface](#) > surface)
- [SegmentHelper::BasicCodedEntry](#) const & [GetAnatomicRegion](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetAnatomicRegion](#) ()
- [BasicCodedEntryVector](#) const & [GetAnatomicRegionModifiers](#) () const
- [BasicCodedEntryVector](#) & [GetAnatomicRegionModifiers](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyCategory](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyCategory](#) ()
- [SegmentHelper::BasicCodedEntry](#) const & [GetPropertyType](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyType](#) ()
- [BasicCodedEntryVector](#) const & [GetPropertyTypeModifiers](#) () const
- [BasicCodedEntryVector](#) & [GetPropertyTypeModifiers](#) ()
- 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](#) const & [GetSurfaces](#) () const
- [SurfaceVector](#) & [GetSurfaces](#) ()
- 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)

## 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

### 10.261.1 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

### 10.261.2 Member Typedef Documentation

#### 10.261.2.1 BasicCodedEntryVector

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcm::Segment::BasicCodedEntryVector
```

#### 10.261.2.2 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

### 10.261.3 Member Enumeration Documentation

#### 10.261.3.1 ALGOType

```
enum gdcm::Segment::ALGOType
```

**Enumerator**

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

**10.261.4 Constructor & Destructor Documentation****10.261.4.1 Segment()**

```
gdcM::Segment::Segment ( )
```

**10.261.4.2 ~Segment()**

```
virtual gdcM::Segment::~~Segment ( ) [virtual]
```

**10.261.5 Member Function Documentation****10.261.5.1 AddSurface()**

```
void gdcM::Segment::AddSurface (
    SmartPointer< Surface > surface )
```

**10.261.5.2 GetALGOType()**

```
static ALGOType gdcM::Segment::GetALGOType (
    const char * type ) [static]
```



### 10.261.5.3 GetALGOTypeString()

```
static const char* gdcm::Segment::GetALGOTypeString (
    ALGOType type ) [static]
```

### 10.261.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const
```

### 10.261.5.5 GetAnatomicRegion() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )
```

### 10.261.5.6 GetAnatomicRegionModifiers() [1/2]

```
BasicCodedEntryVector const& gdcm::Segment::GetAnatomicRegionModifiers ( ) const
```

### 10.261.5.7 GetAnatomicRegionModifiers() [2/2]

```
BasicCodedEntryVector& gdcm::Segment::GetAnatomicRegionModifiers ( )
```

### 10.261.5.8 GetPropertyCategory() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const
```

### 10.261.5.9 GetPropertyCategory() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )
```

**10.261.5.10 GetPropertyType()** [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const
```

**10.261.5.11 GetPropertyType()** [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )
```

**10.261.5.12 GetPropertyTypeModifiers()** [1/2]

```
BasicCodedEntryVector const& gdcm::Segment::GetPropertyTypeModifiers ( ) const
```

**10.261.5.13 GetPropertyTypeModifiers()** [2/2]

```
BasicCodedEntryVector& gdcm::Segment::GetPropertyTypeModifiers ( )
```

**10.261.5.14 GetSegmentAlgorithmName()**

```
const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const
```

**10.261.5.15 GetSegmentAlgorithmType()**

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const
```

**10.261.5.16 GetSegmentDescription()**

```
const char* gdcm::Segment::GetSegmentDescription ( ) const
```

**10.261.5.17 GetSegmentLabel()**

```
const char* gdcm::Segment::GetSegmentLabel ( ) const
```

**10.261.5.18 GetSegmentNumber()**

```
unsigned short gdcm::Segment::GetSegmentNumber ( ) const
```

**10.261.5.19 GetSurface()**

```
SmartPointer< Surface > gdcm::Segment::GetSurface (
    const unsigned int idx = 0 ) const
```

**10.261.5.20 GetSurfaceCount()**

```
unsigned long gdcm::Segment::GetSurfaceCount ( )
```

**10.261.5.21 GetSurfaces()** [1/2]

```
SurfaceVector const& gdcm::Segment::GetSurfaces ( ) const
```

**10.261.5.22 GetSurfaces()** [2/2]

```
SurfaceVector& gdcm::Segment::GetSurfaces ( )
```

**10.261.5.23 SetAnatomicRegion()**

```
void gdcm::Segment::SetAnatomicRegion (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.261.5.24 SetAnatomicRegionModifiers()**

```
void gdcM::Segment::SetAnatomicRegionModifiers (
    BasicCodedEntryVector const & BSEV )
```

**10.261.5.25 SetPropertyCategory()**

```
void gdcM::Segment::SetPropertyCategory (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.261.5.26 SetPropertyType()**

```
void gdcM::Segment::SetPropertyType (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.261.5.27 SetPropertyTypeModifiers()**

```
void gdcM::Segment::SetPropertyTypeModifiers (
    BasicCodedEntryVector const & BSEV )
```

**10.261.5.28 SetSegmentAlgorithmName()**

```
void gdcM::Segment::SetSegmentAlgorithmName (
    const char * name )
```

**10.261.5.29 SetSegmentAlgorithmType()** [1/2]

```
void gdcM::Segment::SetSegmentAlgorithmType (
    ALGOType type )
```

**10.261.5.30 SetSegmentAlgorithmType()** [2/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
    const char * typeStr )
```

**10.261.5.31 SetSegmentDescription()**

```
void gdcm::Segment::SetSegmentDescription (
    const char * description )
```

**10.261.5.32 SetSegmentLabel()**

```
void gdcm::Segment::SetSegmentLabel (
    const char * label )
```

**10.261.5.33 SetSegmentNumber()**

```
void gdcm::Segment::SetSegmentNumber (
    const unsigned short num )
```

**10.261.5.34 SetSurfaceCount()**

```
void gdcm::Segment::SetSurfaceCount (
    const unsigned long nb )
```

**10.261.6 Member Data Documentation****10.261.6.1 AnatomicRegion**

[SegmentHelper::BasicCodedEntry](#) gdcm::Segment::AnatomicRegion [protected]

#### 10.261.6.2 AnatomicRegionModifiers

`BasicCodedEntryVector` `gdcM::Segment::AnatomicRegionModifiers` [protected]

#### 10.261.6.3 PropertyCategory

`SegmentHelper::BasicCodedEntry` `gdcM::Segment::PropertyCategory` [protected]

#### 10.261.6.4 PropertyType

`SegmentHelper::BasicCodedEntry` `gdcM::Segment::PropertyType` [protected]

#### 10.261.6.5 PropertyTypeModifiers

`BasicCodedEntryVector` `gdcM::Segment::PropertyTypeModifiers` [protected]

#### 10.261.6.6 SegmentAlgorithmName

`std::string` `gdcM::Segment::SegmentAlgorithmName` [protected]

#### 10.261.6.7 SegmentAlgorithmType

`ALGOType` `gdcM::Segment::SegmentAlgorithmType` [protected]

#### 10.261.6.8 SegmentDescription

`std::string` `gdcM::Segment::SegmentDescription` [protected]

### 10.261.6.9 SegmentLabel

```
std::string gdcm::Segment::SegmentLabel [protected]
```

### 10.261.6.10 SegmentNumber

```
unsigned short gdcm::Segment::SegmentNumber [protected]
```

### 10.261.6.11 SurfaceCount

```
unsigned long gdcm::Segment::SurfaceCount [protected]
```

### 10.261.6.12 Surfaces

```
SurfaceVector gdcm::Segment::Surfaces [protected]
```

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

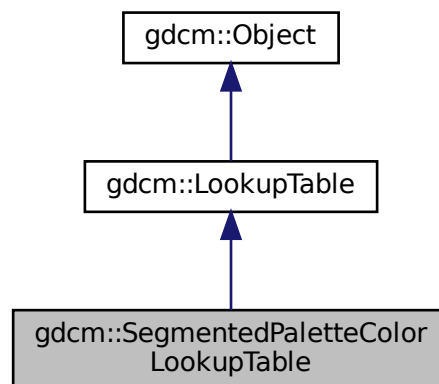
- [gdcmSegment.h](#)

## 10.262 gdcm::SegmentedPaletteColorLookupTable Class Reference

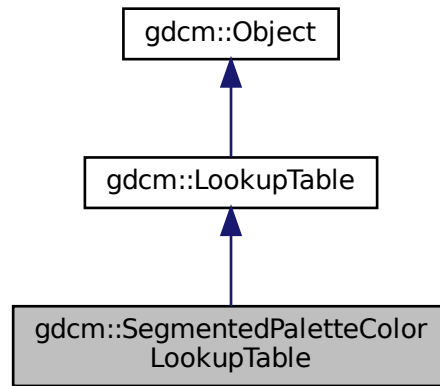
[SegmentedPaletteColorLookupTable](#) class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for `gdcm::SegmentedPaletteColorLookupTable`:



### Public Member Functions

- [SegmentedPaletteColorLookupTable](#) ()
- [~SegmentedPaletteColorLookupTable](#) ()
- void [Print](#) (std::ostream &) const
- void [SetLUT](#) ([LookupTableType](#) type, const unsigned char \*array, unsigned int length)  
*Initialize a [SegmentedPaletteColorLookupTable](#).*

### Additional Inherited Members

#### 10.262.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

#### 10.262.2 Constructor & Destructor Documentation

##### 10.262.2.1 [SegmentedPaletteColorLookupTable](#)()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )
```



### 10.262.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ( )
```

## 10.262.3 Member Function Documentation

### 10.262.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (
    std::ostream & ) const [inline], [virtual]
```

Reimplemented from [gdcm::LookupTable](#).

### 10.262.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (
    LookupTableType type,
    const unsigned char * array,
    unsigned int length ) [virtual]
```

Initialize a [SegmentedPaletteColorLookupTable](#).

Reimplemented from [gdcm::LookupTable](#).

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

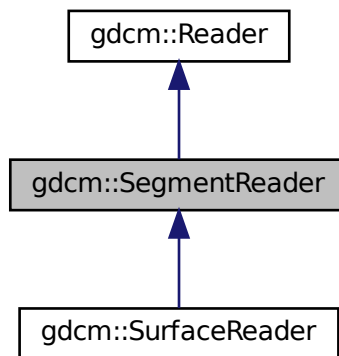
- [gdcmSegmentedPaletteColorLookupTable.h](#)

## 10.263 gdcM::SegmentReader Class Reference

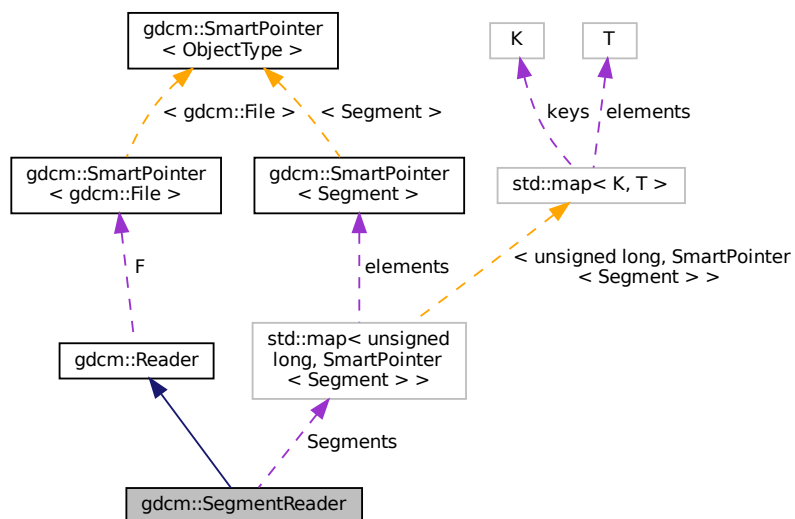
This class defines a segment reader.

```
#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](#) ()
- virtual [~SegmentReader](#) ()
- const [SegmentVector](#) [GetSegments](#) () const
- [SegmentVector](#) [GetSegments](#) ()
- virtual bool [Read](#) ()  
*Read.*

## 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 Attributes

- [SegmentMap](#) [Segments](#)

### 10.263.1 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

### 10.263.2 Member Typedef Documentation

### 10.263.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcM::SegmentReader::SegmentMap [protected]
```

### 10.263.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcM::SegmentReader::SegmentVector
```

## 10.263.3 Constructor & Destructor Documentation

### 10.263.3.1 SegmentReader()

```
gdcM::SegmentReader::SegmentReader ( )
```

### 10.263.3.2 ~SegmentReader()

```
virtual gdcM::SegmentReader::~~SegmentReader ( ) [virtual]
```

## 10.263.4 Member Function Documentation

### 10.263.4.1 GetSegments() [1/2]

```
const SegmentVector gdcM::SegmentReader::GetSegments ( ) const
```

### 10.263.4.2 GetSegments() [2/2]

```
SegmentVector gdcM::SegmentReader::GetSegments ( )
```

#### 10.263.4.3 Read()

```
virtual bool gdcm::SegmentReader::Read ( ) [virtual]
```

Read.

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::SurfaceReader](#).

#### 10.263.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
    const Item & segmentItem,
    const unsigned int idx ) [protected]
```

#### 10.263.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments ( ) [protected]
```

### 10.263.5 Member Data Documentation

#### 10.263.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments [protected]
```

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

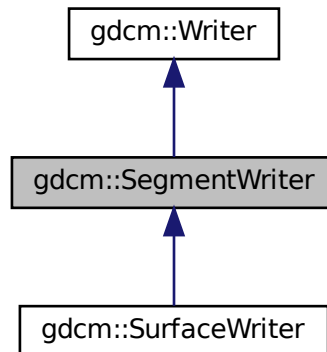
- [gdcmSegmentReader.h](#)

## 10.264 gdcm::SegmentWriter Class Reference

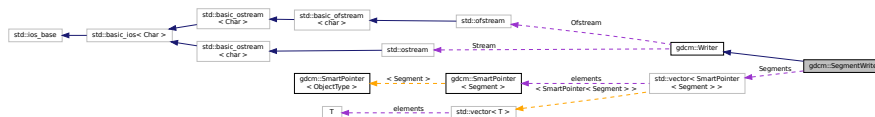
This class defines a segment writer.

```
#include <gdcmSegmentWriter.h>
```

Inheritance diagram for gdcm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



### Public Types

- typedef `std::vector< SmartPointer<Segment> >` [SegmentVector](#)

### Public Member Functions

- [SegmentWriter](#) ()
- virtual `~SegmentWriter` ()
- void [AddSegment](#) ([SmartPointer<Segment>](#) segment)
- unsigned int [GetNumberOfSegments](#) () const
- [SmartPointer<Segment>](#) [GetSegment](#) (const unsigned int idx=0) const
- const [SegmentVector](#) & [GetSegments](#) () const
- [SegmentVector](#) & [GetSegments](#) ()
- void [SetNumberOfSegments](#) (const unsigned int size)
- void [SetSegments](#) ([SegmentVector](#) &segments)
- bool [Write](#) ()

*Write.*

## Protected Member Functions

- bool [PrepareWrite](#) ()

## Protected Attributes

- [SegmentVector](#) [Segments](#)

### 10.264.1 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

### 10.264.2 Member Typedef Documentation

#### 10.264.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

### 10.264.3 Constructor & Destructor Documentation

#### 10.264.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ( )
```

#### 10.264.3.2 ~SegmentWriter()

```
virtual gdcm::SegmentWriter::~SegmentWriter ( ) [virtual]
```

## 10.264.4 Member Function Documentation

### 10.264.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
    SmartPointer< Segment > segment )
```

### 10.264.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const
```

### 10.264.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
    const unsigned int idx = 0 ) const
```

### 10.264.4.4 GetSegments() [1/2]

```
const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const
```

### 10.264.4.5 GetSegments() [2/2]

```
SegmentVector& gdcm::SegmentWriter::GetSegments ( )
```

### 10.264.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]
```



#### 10.264.4.7 SetNumberOfSegments()

```
void gdcM::SegmentWriter::SetNumberOfSegments (
    const unsigned int size )
```

#### 10.264.4.8 SetSegments()

```
void gdcM::SegmentWriter::SetSegments (
    SegmentVector & segments )
```

#### 10.264.4.9 Write()

```
bool gdcM::SegmentWriter::Write ( ) [virtual]
```

Write.

Reimplemented from [gdcM::Writer](#).

Reimplemented in [gdcM::SurfaceWriter](#).

### 10.264.5 Member Data Documentation

#### 10.264.5.1 Segments

```
SegmentVector gdcM::SegmentWriter::Segments [protected]
```

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

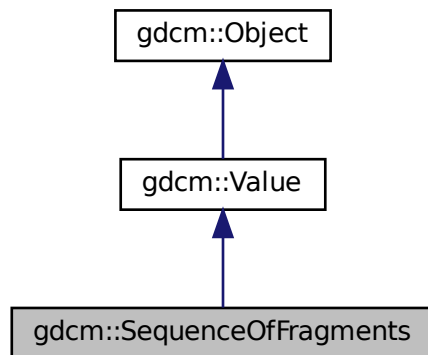
- [gdcMSegmentWriter.h](#)

## 10.265 gdcmm::SequenceOfFragments Class Reference

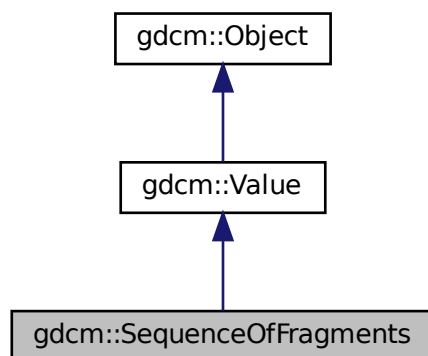
Class to represent a Sequence Of Fragments.

```
#include <gdcmmSequenceOfFragments.h>
```

Inheritance diagram for gdcmm::SequenceOfFragments:



Collaboration diagram for gdcmm::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](#) ()  
*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  
*Returns the SQ length, as read from disk.*
- [SizeType GetNumberOfFragments](#) () const
- const [BasicOffsetTable](#) & [GetTable](#) () const
- [BasicOffsetTable](#) & [GetTable](#) ()
- bool [operator==](#) (const [Value](#) &val) const
- void [Print](#) (std::ostream &os) const
- 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)  
*Sets the actual SQ length.*
- template<typename TSwap >  
std::ostream const & [Write](#) (std::ostream &os) const
- bool [WriteBuffer](#) (std::ostream &os) const

## Static Public Member Functions

- static [SmartPointer](#)< [SequenceOfFragments](#) > [New](#) ()

## Additional Inherited Members

### 10.265.1 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:

[FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), and [GetJPEGSamplePrecision.cxx](#).

## 10.265.2 Member Typedef Documentation

### 10.265.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

### 10.265.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

### 10.265.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

### 10.265.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

## 10.265.3 Constructor & Destructor Documentation

### 10.265.3.1 SequenceOfFragments()

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

constructor (UndefinedLength by default)

## 10.265.4 Member Function Documentation

#### 10.265.4.1 AddFragment()

```
void gdcm::SequenceOfFragments::AddFragment (
    Fragment const & item )
```

Appends a [Fragment](#) to the already added ones.

Examples:

[FixBrokenJ2K.cxx](#).

#### 10.265.4.2 Begin() [1/2]

```
Iterator gdcm::SequenceOfFragments::Begin ( ) [inline]
```

#### 10.265.4.3 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::Begin ( ) const [inline]
```

#### 10.265.4.4 Clear()

```
void gdcm::SequenceOfFragments::Clear ( ) [virtual]
```

Clear.

Implements [gdcm::Value](#).

#### 10.265.4.5 ComputeByteLength()

```
unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const
```

#### 10.265.4.6 ComputeLength()

```
VL gdcm::SequenceOfFragments::ComputeLength ( ) const
```

**10.265.4.7 End()** [1/2]

```
Iterator gdcM::SequenceOfFragments::End ( ) [inline]
```

**10.265.4.8 End()** [2/2]

```
ConstIterator gdcM::SequenceOfFragments::End ( ) const [inline]
```

**10.265.4.9 GetBuffer()**

```
bool gdcM::SequenceOfFragments::GetBuffer (
    char * buffer,
    unsigned long length ) const
```

**10.265.4.10 GetFragBuffer()**

```
bool gdcM::SequenceOfFragments::GetFragBuffer (
    unsigned int fragNb,
    char * buffer,
    unsigned long & length ) const
```

**10.265.4.11 GetFragment()**

```
const Fragment& gdcM::SequenceOfFragments::GetFragment (
    SizeType num ) const
```

**Examples:**

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

#### 10.265.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

#### 10.265.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const
```

Examples:

[FixJAIBugJPEGLS.cxx](#).

#### 10.265.4.14 GetTable() [1/2]

```
const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const [inline]
```

#### 10.265.4.15 GetTable() [2/2]

```
BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) [inline]
```

#### 10.265.4.16 New()

```
static SmartPointer<SequenceOfFragments> gdcm::SequenceOfFragments::New ( ) [inline], [static]
```

#### 10.265.4.17 operator==()

```
bool gdcm::SequenceOfFragments::operator== (
    const Value & val ) const [inline], [virtual]
```

Implements [gdcm::Value](#).

**10.265.4.18 Print()**

```
void gdcM::SequenceOfFragments::Print (
    std::ostream & os ) const [inline], [virtual]
```

Reimplemented from [gdcM::Object](#).

**10.265.4.19 Read()**

```
template<typename TSwap >
std::istream& gdcM::SequenceOfFragments::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

**10.265.4.20 ReadPreValue()**

```
template<typename TSwap >
std::istream& gdcM::SequenceOfFragments::ReadPreValue (
    std::istream & is ) [inline]
```

References [gdcMDebugMacro](#).

**10.265.4.21 ReadValue()**

```
template<typename TSwap >
std::istream& gdcM::SequenceOfFragments::ReadValue (
    std::istream & is,
    bool ) [inline]
```

References [gdcMAssertAlwaysMacro](#), [gdcMDebugMacro](#), [gdcMWarningMacro](#), [gdcM::Tag::GetElement\(\)](#), [gdcM::Tag::GetGroup\(\)](#), [gdcM::ByteValue::GetLength\(\)](#), [gdcM::ByteValue::GetPointer\(\)](#), [gdcM::DataElement::GetTag\(\)](#), [gdcM::DataElement::GetVL\(\)](#), [gdcM::Fragment::Read\(\)](#), [gdcM::Fragment::ReadBacktrack\(\)](#), and [gdcM::Exception::what\(\)](#).

**10.265.4.22 SetLength()**

```
void gdcM::SequenceOfFragments::SetLength (
    VL length ) [inline], [virtual]
```

Sets the actual SQ length.

Implements [gdcM::Value](#).



#### 10.265.4.23 Write()

```
template<typename TSwap >
std::ostream const& gdcm::SequenceOfFragments::Write (
    std::ostream & os ) const [inline]
```

References `gdcm::VL::Write()`.

#### 10.265.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
    std::ostream & os ) const
```

#### Examples:

[GetJPEGSamplePrecision.cxx](#).

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

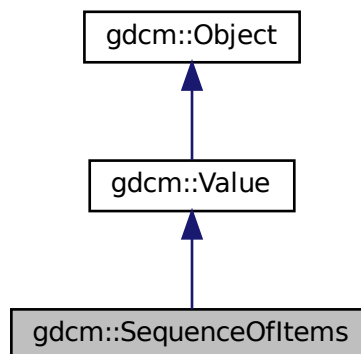
- [gdcmSequenceOfFragments.h](#)

## 10.266 gdcm::SequenceOfItems Class Reference

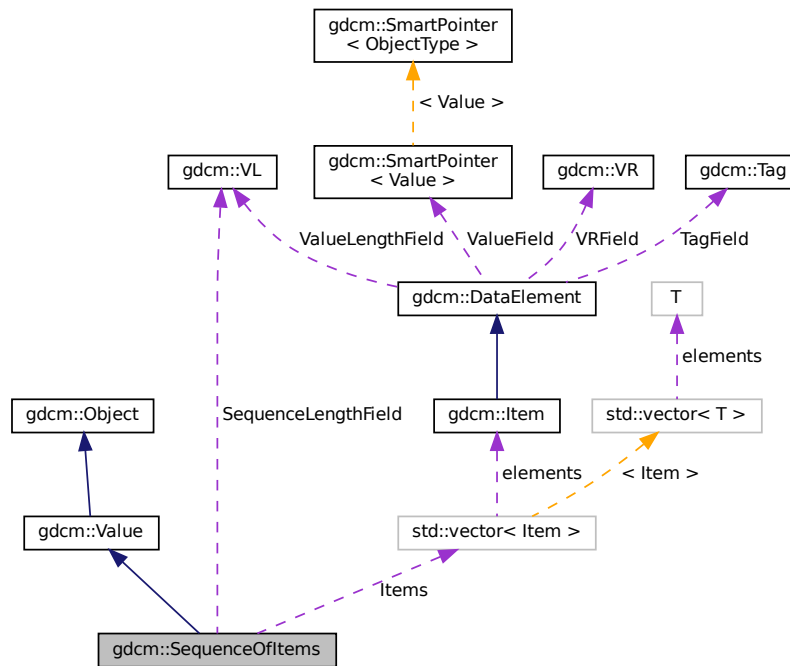
Class to represent a Sequence Of Items.

```
#include <gdcmSequenceOfItems.h>
```

Inheritance diagram for `gdcm::SequenceOfItems`:



Collaboration diagram for `gdcM::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](#) ()  
*remove all items within the sequence*
- `template<typename TDE >`  
`VL` [ComputeLength](#) () const

- [Iterator End](#) ()
- [ConstIterator End](#) () const
- bool [FindDataElement](#) (const [Tag](#) &t) const
- const [Item](#) & [GetItem](#) ([SizeType](#) position) const
- [Item](#) & [GetItem](#) ([SizeType](#) position)
- [VL GetLength](#) () const  
*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
- void [Print](#) (std::ostream &os) const
- template<typename TDE , typename TSwap >  
std::istream & [Read](#) (std::istream &is, bool readvalues=true)
- bool [RemoveItemByIndex](#) (const [SizeType](#) index)
- void [SetLength](#) ([VL](#) length)  
*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

## 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

### 10.266.1 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](#).

**10.266.2 Member Typedef Documentation****10.266.2.1 ConstIterator**

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

**10.266.2.2 ItemVector**

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

**10.266.2.3 Iterator**

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

**10.266.2.4 SizeType**

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

**10.266.3 Constructor & Destructor Documentation**

### 10.266.3.1 SequenceOfItems()

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

constructor (UndefinedLength by default)

## 10.266.4 Member Function Documentation

### 10.266.4.1 AddItem()

```
void gdcm::SequenceOfItems::AddItem (
    Item const & item )
```

Appends an [Item](#) to the already added ones.

Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

### 10.266.4.2 AddNewUndefinedLengthItem()

```
Item& gdcm::SequenceOfItems::AddNewUndefinedLengthItem ( )
```

Appends an [Item](#) to the already added ones.

### 10.266.4.3 Begin() [1/2]

```
Iterator gdcm::SequenceOfItems::Begin ( ) [inline]
```

### 10.266.4.4 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfItems::Begin ( ) const [inline]
```

#### 10.266.4.5 Clear()

```
void gdcM::SequenceOfItems::Clear ( ) [virtual]
```

remove all items within the sequence

Implements [gdcM::Value](#).

#### 10.266.4.6 ComputeLength()

```
template<typename TDE >  
VL gdcM::SequenceOfItems::ComputeLength ( ) const
```

#### 10.266.4.7 End() [1/2]

```
Iterator gdcM::SequenceOfItems::End ( ) [inline]
```

#### 10.266.4.8 End() [2/2]

```
ConstIterator gdcM::SequenceOfItems::End ( ) const [inline]
```

#### 10.266.4.9 FindDataElement()

```
bool gdcM::SequenceOfItems::FindDataElement (   
    const Tag & t ) const
```

#### 10.266.4.10 GetItem() [1/2]

```
const Item& gdcM::SequenceOfItems::GetItem (   
    SizeType position ) const
```

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcMrtionplan.cxx](#), [gdcMrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

#### 10.266.4.11 GetItem() [2/2]

```
Item& gdcm::SequenceOfItems::GetItem (
    SizeType position )
```

#### 10.266.4.12 GetLength()

```
VL gdcm::SequenceOfItems::GetLength ( ) const [inline], [virtual]
```

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

#### 10.266.4.13 GetNumberOfItems()

```
SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const [inline]
```

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DumpToshibaDTI.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrionplan.cxx](#), [gdcmrtpplan.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

#### 10.266.4.14 IsEmpty()

```
bool gdcm::SequenceOfItems::IsEmpty ( ) const [inline]
```

#### 10.266.4.15 IsUndefinedLength()

```
bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]
```

return if [Value](#) Length if of undefined length

**10.266.4.16 New()**

```
static SmartPointer<SequenceOfItems> gdcM::SequenceOfItems::New ( ) [inline], [static]
```

Examples:

[NewSequence.cs](#).

**10.266.4.17 operator=()**

```
SequenceOfItems& gdcM::SequenceOfItems::operator= (
    const SequenceOfItems & val ) [inline]
```

References [Items](#), and [SequenceLengthField](#).

**10.266.4.18 operator==()**

```
bool gdcM::SequenceOfItems::operator== (
    const Value & val ) const [inline], [virtual]
```

Implements [gdcM::Value](#).

References [Items](#), and [SequenceLengthField](#).

**10.266.4.19 Print()**

```
void gdcM::SequenceOfItems::Print (
    std::ostream & os ) const [inline], [virtual]
```

Reimplemented from [gdcM::Object](#).



#### 10.266.4.20 Read()

```
template<typename TDE , typename TSwap >
std::istream& gdcm::SequenceOfItems::Read (
    std::istream & is,
    bool readvalues = true ) [inline]
```

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

References [gdcm::Item::Clear\(\)](#), [gdcmDebugMacro](#), [gdcmWarningMacro](#), [gdcm::Exception::GetDescription\(\)](#), [gdcm::Item::GetNestedDataSet\(\)](#), [gdcm::DataElement::GetTag\(\)](#), [gdcm::DataElement::GetVL\(\)](#), [gdcm::Item::Read\(\)](#), and [gdcm::DataSet::Size\(\)](#).

#### 10.266.4.21 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

#### 10.266.4.22 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
    VL length ) [inline], [virtual]
```

Sets the actual SQ length.

Implements [gdcm::Value](#).

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

#### 10.266.4.23 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ( )
```

Properly set the Sequence of [Item](#) to be undefined length.

Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

#### 10.266.4.24 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
    SizeType n ) [inline]
```

#### 10.266.4.25 Write()

```
template<typename TDE , typename TSwap >
std::ostream const& gdcm::SequenceOfItems::Write (
    std::ostream & os ) const [inline]
```

References `gdcm::VL::Write()`, and `gdcm::Tag::Write()`.

### 10.266.5 Member Data Documentation

#### 10.266.5.1 Items

```
ItemVector gdcm::SequenceOfItems::Items
```

Vector of Sequence Items.

Referenced by `operator=()`, and `operator==()`.

#### 10.266.5.2 SequenceLengthField

```
VL gdcm::SequenceOfItems::SequenceLengthField
```

Total length of the Sequence (or 0xffffffff) if undefined.

Referenced by `operator=()`, and `operator==()`.

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

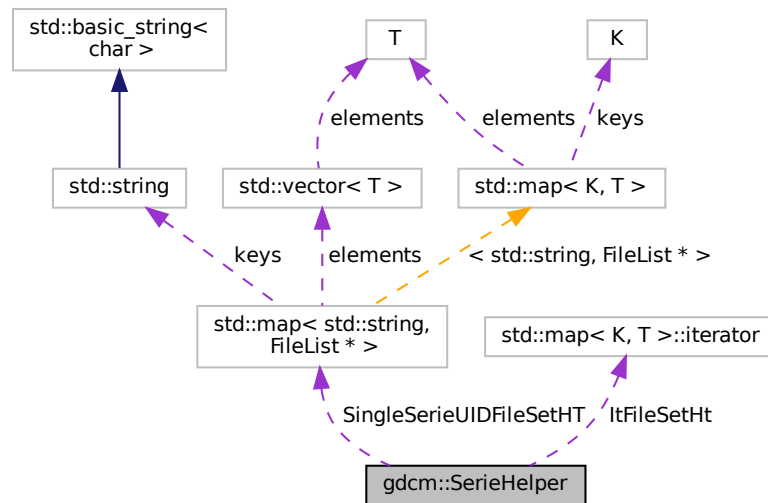
- [gdcmSequenceOfItems.h](#)

## 10.267 gdcm::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 disapear soon, you've been warned.

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper:



### Classes

- struct [Rule](#)

### 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

- 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 [ImagePositionPatientOrdering](#) ([FileList](#) \*fileSet)
- bool [UserOrdering](#) ([FileList](#) \*fileSet)

## Protected Attributes

- [SingleSerieUIDFileSetmap::iterator](#) [ItFileSetHt](#)
- [SingleSerieUIDFileSetmap](#) [SingleSerieUIDFileSetHT](#)

### 10.267.1 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](#)

### 10.267.2 Member Typedef Documentation

#### 10.267.2.1 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]
```

#### 10.267.2.2 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]
```

### 10.267.3 Constructor & Destructor Documentation

### 10.267.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ( )
```

### 10.267.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~~SerieHelper ( )
```

## 10.267.4 Member Function Documentation

### 10.267.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
    FileWithName & header ) [protected]
```

### 10.267.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
    std::string const & filename ) [protected]
```

### 10.267.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
    const std::string & tag )
```

### 10.267.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
    uint16_t group,
    uint16_t elem,
    std::string const & value,
    int op )
```

**10.267.4.5 AddRestriction()** [3/3]

```
void gdcm::SerieHelper::AddRestriction (
    const Tag & tag ) [protected]
```

**10.267.4.6 Clear()**

```
void gdcm::SerieHelper::Clear ( )
```

**10.267.4.7 CreateDefaultUniqueSeriesIdentifier()**

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
```

**10.267.4.8 CreateUniqueSeriesIdentifier()**

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
    File * inFile )
```

**10.267.4.9 FileNameOrdering()**

```
bool gdcm::SerieHelper::FileNameOrdering (
    FileList * fileList ) [protected]
```

**10.267.4.10 GetFirstSingleSerieUIDFileSet()**

```
FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
```

**10.267.4.11 GetNextSingleSerieUIDFileSet()**

```
FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
```

#### 10.267.4.12 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
    FileList * fileSet ) [protected]
```

#### 10.267.4.13 OrderFileList()

```
void gdcm::SerieHelper::OrderFileList (
    FileList * fileSet )
```

#### 10.267.4.14 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
    std::string const & dir,
    bool recursive = false )
```

#### 10.267.4.15 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
    int ) [inline]
```

#### 10.267.4.16 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
    bool useSeriesDetails )
```

#### 10.267.4.17 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
    FileList * fileSet ) [protected]
```

### 10.267.5 Member Data Documentation

### 10.267.5.1 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcM::SerieHelper::ItFileSetHt [protected]
```

### 10.267.5.2 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcM::SerieHelper::SingleSerieUIDFileSetHT [protected]
```

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

- [gdcMSerieHelper.h](#)

## 10.268 gdcM::Series Class Reference

[Series.](#)

```
#include <gdcMSeries.h>
```

### Public Member Functions

- [Series](#) ()

### 10.268.1 Detailed Description

[Series.](#)

### 10.268.2 Constructor & Destructor Documentation

#### 10.268.2.1 Series()

```
gdcM::Series::Series ( ) [inline]
```

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

- [gdcMSeries.h](#)



## 10.269 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

### 10.269.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

### 10.269.2 Constructor & Destructor Documentation

#### 10.269.2.1 ServiceClassApplicationInformation()

```
gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )
```

### 10.269.3 Member Function Documentation

#### 10.269.3.1 Print()

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

#### 10.269.3.2 Read()

```
std::istream& gdcm::network::ServiceClassApplicationInformation::Read (
    std::istream & is )
```

### 10.269.3.3 SetTuple()

```
void gdcmm::network::ServiceClassApplicationInformation::SetTuple (
    uint8_t levelofsupport,
    uint8_t levelofdigitalsig,
    uint8_t elementcoercion )
```

### 10.269.3.4 Size()

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

### 10.269.3.5 Write()

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

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

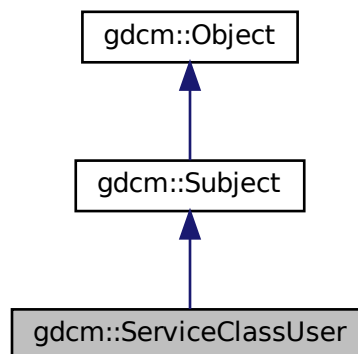
- [gdcmmServiceClassApplicationInformation.h](#)

## 10.270 gdcmm::ServiceClassUser Class Reference

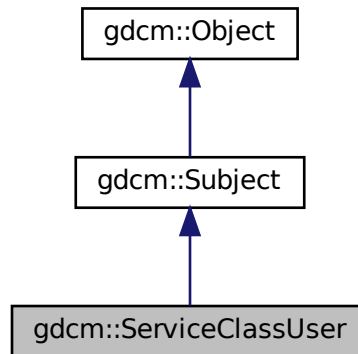
[ServiceClassUser](#).

```
#include <gdcmmServiceClassUser.h>
```

Inheritance diagram for gdcmm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



## Public Member Functions

- [ServiceClassUser](#) ()
- [~ServiceClassUser](#) ()
- 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.*
- 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](#) ([File](#) const &file)
- bool [SendStore](#) ([DataSet](#) const &ds)  
*Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.*
- 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.*

## Static Public Member Functions

- static [SmartPointer](#)< [ServiceClassUser](#) > [New](#) ()  
*for wrapped language: instantiate a reference counted object*

## Additional Inherited Members

### 10.270.1 Detailed Description

[ServiceClassUser](#).

Examples:

[CStoreQtProgress.cxx](#).

### 10.270.2 Constructor & Destructor Documentation

#### 10.270.2.1 ServiceClassUser()

```
gdcmm::ServiceClassUser::ServiceClassUser ( )
```

Construct a SCU with default:

- hostname = localhost
- port = 104

### 10.270.2.2 ~ServiceClassUser()

```
gdcmm::ServiceClassUser::~~ServiceClassUser ( )
```

## 10.270.3 Member Function Documentation

### 10.270.3.1 GetAETitle()

```
const char* gdcmm::ServiceClassUser::GetAETitle ( ) const
```

### 10.270.3.2 GetCalledAETitle()

```
const char* gdcmm::ServiceClassUser::GetCalledAETitle ( ) const
```

### 10.270.3.3 GetTimeout()

```
double gdcmm::ServiceClassUser::GetTimeout ( ) const
```

### 10.270.3.4 InitializeConnection()

```
bool gdcmm::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](#).

### 10.270.3.5 IsPresentationContextAccepted()

```
bool gdcM::ServiceClassUser::IsPresentationContextAccepted (
    const PresentationContext & pc ) const
```

Return if the passed in presentation was accepted during association negotiation.

### 10.270.3.6 New()

```
static SmartPointer<ServiceClassUser> gdcM::ServiceClassUser::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

### 10.270.3.7 SendEcho()

```
bool gdcM::ServiceClassUser::SendEcho ( )
```

C-ECHO.

### 10.270.3.8 SendFind()

```
bool gdcM::ServiceClassUser::SendFind (
    const BaseRootQuery * query,
    std::vector< DataSet > & retDatasets )
```

C-FIND a query, return result are in retDatasets.

### 10.270.3.9 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.

**10.270.3.10 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.

**10.270.3.11 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.

**10.270.3.12 SendStore()** [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
    const char * filename )
```

Execute a C-STORE on file on disk, named filename.

Examples:

[CStoreQtProgress.cxx](#).

**10.270.3.13 SendStore()** [2/3]

```
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.

**10.270.3.14 SendStore()** [3/3]

```
bool gdcm::ServiceClassUser::SendStore (
    DataSet const & ds )
```

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

**10.270.3.15 SetAETitle()**

```
void gdcmm::ServiceClassUser::SetAETitle (
    const char * aetitle )
```

set calling ae title

**10.270.3.16 SetCalledAETitle()**

```
void gdcmm::ServiceClassUser::SetCalledAETitle (
    const char * aetitle )
```

set called ae title

**Examples:**

[CStoreQtProgress.cxx](#).

**10.270.3.17 SetHostname()**

```
void gdcmm::ServiceClassUser::SetHostname (
    const char * hostname )
```

Set the name of the called hostname (hostname or IP address)

**Examples:**

[CStoreQtProgress.cxx](#).

**10.270.3.18 SetPort()**

```
void gdcmm::ServiceClassUser::SetPort (
    uint16_t port )
```

Set port of remote host (called application)

**Examples:**

[CStoreQtProgress.cxx](#).



### 10.270.3.19 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)

### 10.270.3.20 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
    std::vector< PresentationContext > const & pcs )
```

Set the Presentation Context used for the Association.

Examples:

[CStoreQtProgress.cxx](#).

### 10.270.3.21 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
    double t )
```

set/get Timeout

Examples:

[CStoreQtProgress.cxx](#).

### 10.270.3.22 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ( )
```

Start the association. Need to call SetPresentationContexts before.

Examples:

[CStoreQtProgress.cxx](#).

### 10.270.3.23 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](#)

## 10.271 gdcmm::SHA1 Class Reference

Class for [SHA1](#).

```
#include <gdcmmSHA1.h>
```

### Public Member Functions

- [SHA1](#) ()
- [~SHA1](#) ()

### 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])

### 10.271.1 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

## 10.271.2 Constructor & Destructor Documentation

### 10.271.2.1 SHA1()

```
gdcm::SHA1::SHA1 ( )
```

### 10.271.2.2 ~SHA1()

```
gdcm::SHA1::~~SHA1 ( )
```

## 10.271.3 Member Function Documentation

### 10.271.3.1 Compute()

```
static bool gdcm::SHA1::Compute (
    const char * buffer,
    unsigned long buf_len,
    char digest_str[20 *2+1] ) [static]
```

### 10.271.3.2 ComputeFile()

```
static bool gdcm::SHA1::ComputeFile (
    const char * filename,
    char digest_str[20 *2+1] ) [static]
```

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

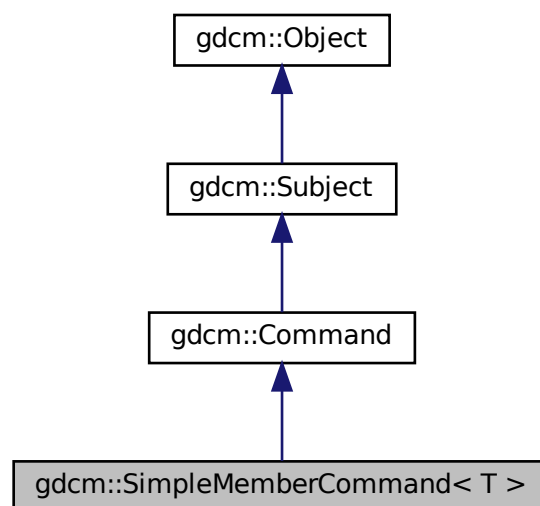
- [gdcmSHA1.h](#)

## 10.272 gdcM::SimpleMemberCommand< T > Class Template Reference

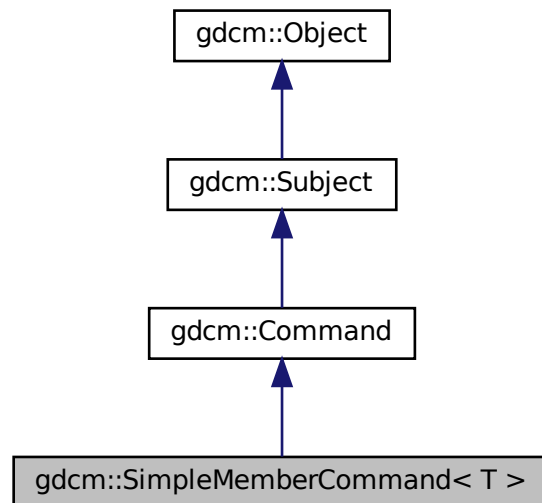
[Command](#) subclass that calls a pointer to a member function.

```
#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

- virtual void [Execute](#) (`Subject *`, const `Event &`)
- virtual void [Execute](#) (const `Subject *`, const `Event &`)
- void [SetCallbackFunction](#) (`T *object`, [TMemberFunctionPointer](#) `memberFunction`)

## Static Public Member Functions

- static [SmartPointer](#)< [SimpleMemberCommand](#) > [New](#) ()

## Protected Member Functions

- [SimpleMemberCommand](#) ()
- virtual [~SimpleMemberCommand](#) ()

## Protected Attributes

- [TMemberFunctionPointer m\\_MemberFunction](#)
- [T \\* m\\_This](#)

### 10.272.1 Detailed Description

```
template<typename T>  
class gdcm::SimpleMemberCommand< T >
```

[Command](#) subclass that calls a pointer to a member function.

[SimpleMemberCommand](#) calls a pointer to a member function with no arguments.

### 10.272.2 Member Typedef Documentation

#### 10.272.2.1 Self

```
template<typename T >  
typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

#### 10.272.2.2 TMemberFunctionPointer

```
template<typename T >  
typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

### 10.272.3 Constructor & Destructor Documentation

#### 10.272.3.1 SimpleMemberCommand()

```
template<typename T >  
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]
```

### 10.272.3.2 ~SimpleMemberCommand()

```
template<typename T >
virtual gdcmm::SimpleMemberCommand< T >::~~SimpleMemberCommand ( ) [inline], [protected], [virtual]
```

## 10.272.4 Member Function Documentation

### 10.272.4.1 Execute() [1/2]

```
template<typename T >
virtual void gdcmm::SimpleMemberCommand< T >::Execute (
    Subject * ,
    const Event & ) [inline], [virtual]
```

Invoke the callback function.

Implements [gdcmm::Command](#).

### 10.272.4.2 Execute() [2/2]

```
template<typename T >
virtual void gdcmm::SimpleMemberCommand< T >::Execute (
    const Subject * caller,
    const Event & event ) [inline], [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 [gdcmm::Command](#).

### 10.272.4.3 New()

```
template<typename T >
static SmartPointer<SimpleMemberCommand> gdcmm::SimpleMemberCommand< T >::New ( ) [inline],
[static]
```

Run-time type information (and related methods). Method for creation through the object factory.

#### 10.272.4.4 SetCallbackFunction()

```
template<typename T >
void gdcM::SimpleMemberCommand< T >::SetCallbackFunction (
    T * object,
    TMemberFunctionPointer memberFunction ) [inline]
```

Specify the callback function.

### 10.272.5 Member Data Documentation

#### 10.272.5.1 m\_MemberFunction

```
template<typename T >
TMemberFunctionPointer gdcM::SimpleMemberCommand< T >::m_MemberFunction [protected]
```

#### 10.272.5.2 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](#)

## 10.273 [gdcM::SimpleSubjectWatcher](#) Class Reference

[SimpleSubjectWatcher](#).

```
#include <gdcMSimpleSubjectWatcher.h>
```

### Public Member Functions

- [SimpleSubjectWatcher](#) ([Subject](#) \*s, const char \*comment="")
- virtual [~SimpleSubjectWatcher](#) ()



## 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](#) ()

### 10.273.1 Detailed Description

[SimpleSubjectWatcher](#).

This is a typical [Subject](#) Watcher class. It will observe all events.

Examples:

[SimpleScanner.cxx](#).

### 10.273.2 Constructor & Destructor Documentation

#### 10.273.2.1 SimpleSubjectWatcher()

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
    Subject * s,
    const char * comment = "" )
```

#### 10.273.2.2 ~SimpleSubjectWatcher()

```
virtual gdcm::SimpleSubjectWatcher::~~SimpleSubjectWatcher ( ) [virtual]
```

### 10.273.3 Member Function Documentation

**10.273.3.1 EndFilter()**

```
virtual void gdcm::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]
```

**10.273.3.2 ShowAbort()**

```
virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]
```

**10.273.3.3 ShowAnonymization()**

```
virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

**10.273.3.4 ShowData()**

```
virtual void gdcm::SimpleSubjectWatcher::ShowData (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

**10.273.3.5 ShowDataSet()**

```
virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

**10.273.3.6 ShowFileName()**

```
virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

**Examples:**

[SimpleScanner.cxx](#).

### 10.273.3.7 ShowIteration()

```
virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]
```

### 10.273.3.8 ShowProgress()

```
virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
    Subject * caller,
    const Event & evt ) [protected], [virtual]
```

### 10.273.3.9 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]
```

### 10.273.3.10 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff ( ) [protected]
```

### 10.273.3.11 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn ( ) [protected]
```

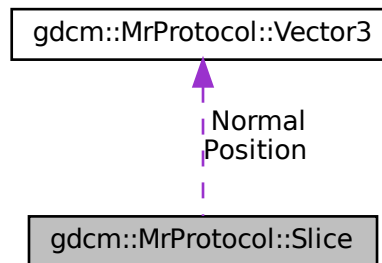
The documentation for this class was generated from the following file:

- [gdcmSimpleSubjectWatcher.h](#)

## 10.274 gdcM::MrProtocol::Slice Struct Reference

```
#include <gdcMMrProtocol.h>
```

Collaboration diagram for gdcM::MrProtocol::Slice:



### Public Attributes

- [Vector3 Normal](#)
- [Vector3 Position](#)

### 10.274.1 Member Data Documentation

#### 10.274.1.1 Normal

```
Vector3 gdcM::MrProtocol::Slice::Normal
```

#### 10.274.1.2 Position

```
Vector3 gdcM::MrProtocol::Slice::Position
```

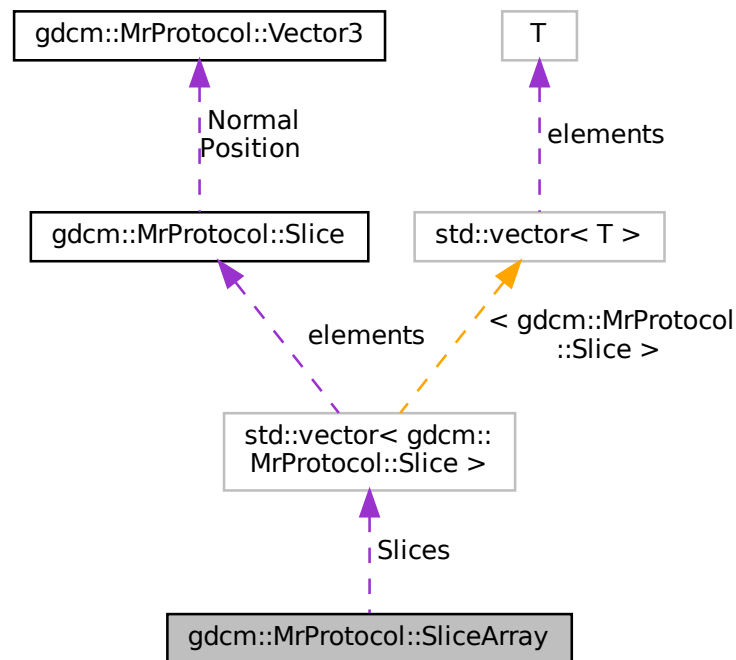
The documentation for this struct was generated from the following file:

- [gdcMMrProtocol.h](#)

## 10.275 gdcm::MrProtocol::SliceArray Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::SliceArray:



### Public Attributes

- `std::vector< Slice > Slices`

### 10.275.1 Member Data Documentation

#### 10.275.1.1 Slices

```
std::vector< Slice > gdcm::MrProtocol::SliceArray::Slices
```

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

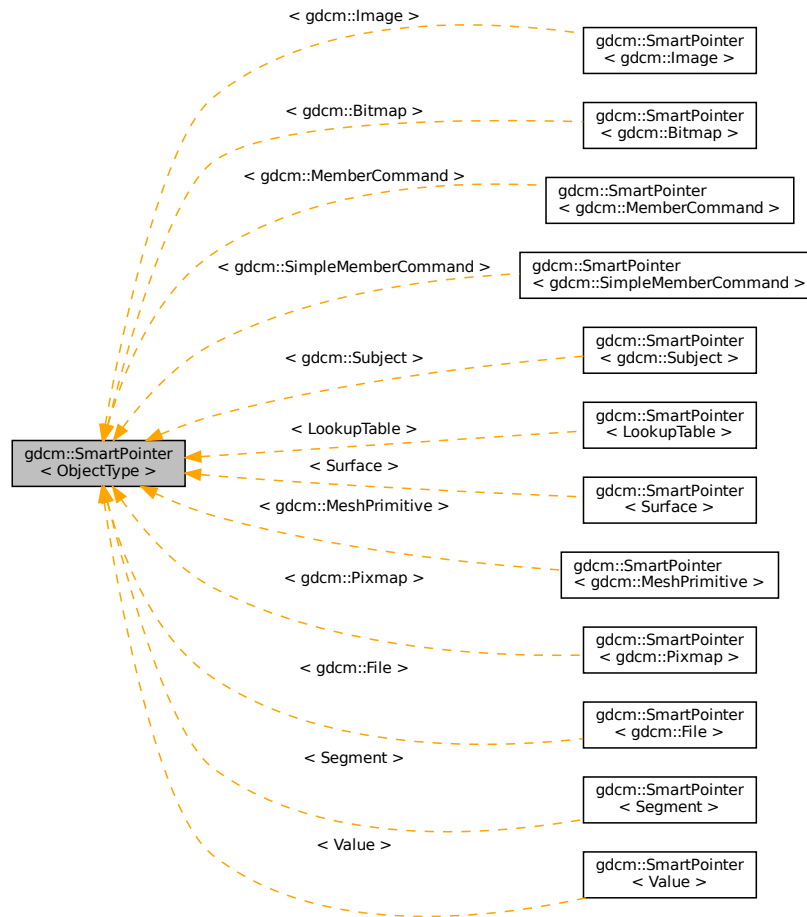
- [gdcmMrProtocol.h](#)

## 10.276 `gdcm::SmartPointer< ObjectType >` Class Template Reference

Class for Smart Pointer.

```
#include <gdcmObject.h>
```

Inheritance diagram for `gdcm::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= (SmartPointer const &r)`  
*Overload operator assignment.*
- `SmartPointer & operator= (ObjectType *r)`  
*Overload operator assignment.*
- `SmartPointer & operator= (ObjectType const &r)`

### 10.276.1 Detailed Description

```
template<class ObjectType>
class gdcmm::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of `gdcmm::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/smarty.htm>

and `itk::SmartPointer`

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [CStoreQtProgress.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECH←O.cxx](#), [DumpToshibaDTI.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLong←Seqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), and [SimpleScanner.cxx](#).

### 10.276.2 Constructor & Destructor Documentation

**10.276.2.1 SmartPointer()** [1/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]
```

**10.276.2.2 SmartPointer()** [2/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
    const SmartPointer< ObjectType > & p ) [inline]
```

**10.276.2.3 SmartPointer()** [3/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType * p ) [inline]
```

**10.276.2.4 SmartPointer()** [4/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
    ObjectType const & p ) [inline]
```

**10.276.2.5 ~SmartPointer()**

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::~SmartPointer ( ) [inline]
```

**10.276.3 Member Function Documentation**



### 10.276.3.1 GetPointer()

```
template<class ObjectType>
ObjectType* gdcmm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]
```

Explicit function to retrieve the pointer.

### 10.276.3.2 operator ObjectType \*()

```
template<class ObjectType>
gdcmm::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]
```

Return pointer to object.

### 10.276.3.3 operator\*()

```
template<class ObjectType>
ObjectType& gdcmm::SmartPointer< ObjectType >::operator* ( ) const [inline]
```

### 10.276.3.4 operator->()

```
template<class ObjectType>
ObjectType* gdcmm::SmartPointer< ObjectType >::operator-> ( ) const [inline]
```

Overload operator ->

### 10.276.3.5 operator=() [1/3]

```
template<class ObjectType>
SmartPointer& gdcmm::SmartPointer< ObjectType >::operator= (
    SmartPointer< ObjectType > const & r ) [inline]
```

Overload operator assignment.

Referenced by gdcmm::SmartPointer< Value >::operator=().

**10.276.3.6 operator=()** [2/3]

```
template<class ObjectType>
SmartPointer& gdcM::SmartPointer< ObjectType >::operator= (
    ObjectType * r ) [inline]
```

Overload operator assignment.

**10.276.3.7 operator=()** [3/3]

```
template<class ObjectType>
SmartPointer& gdcM::SmartPointer< ObjectType >::operator= (
    ObjectType const & r ) [inline]
```

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

- [gdcMObject.h](#)
- [gdcMSmartPointer.h](#)

**10.277 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

**10.277.1 Detailed Description**

[SOPClassExtendedNegociationSub](#).

PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

## 10.277.2 Constructor & Destructor Documentation

### 10.277.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )
```

## 10.277.3 Member Function Documentation

### 10.277.3.1 Print()

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

### 10.277.3.2 Read()

```
std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read (
    std::istream & is )
```

### 10.277.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (
    const char * uid,
    uint8_t levelofsupport = 3,
    uint8_t levelofdigitalsig = 0,
    uint8_t elementcoercion = 2 )
```

### 10.277.3.4 Size()

```
size_t gdcm::network::SOPClassExtendedNegociationSub::Size ( ) const
```

### 10.277.3.5 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](#)

## 10.278 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](#) ()

### 10.278.1 Detailed Description

Class convert a class SOP Class UID into [IOD](#).

Reference PS 3.4 [Table](#) B.5-1 STANDARD SOP CLASSES

### 10.278.2 Member Typedef Documentation

#### 10.278.2.1 const

```
typedef const char* gdcm::SOPClassUIDToIOD::const (SOPClassUIDToIODType) [2]
```

### 10.278.3 Member Function Documentation

#### 10.278.3.1 GetIOD()

```
static 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](#).

#### 10.278.3.2 GetIODFromSOPClassUID()

```
static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
    const char * sopclassuid ) [static]
```

#### 10.278.3.3 GetNumberOfSOPClassToIOD()

```
static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]
```

Return the number of SOP Class UID listed internally.

#### 10.278.3.4 GetSOPClassUIDFromIOD()

```
static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
    const char * iod ) [static]
```

#### 10.278.3.5 GetSOPClassUIDToIOD()

```
static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
    unsigned int i ) [static]
```

### 10.278.3.6 GetSOPClassUIDToIODs()

```
static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]
```

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

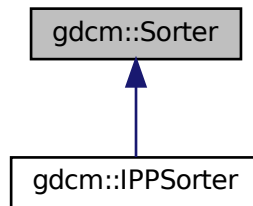
- [gdcmSOPClassUIDToIOD.h](#)

## 10.279 gdcm::Sorter Class Reference

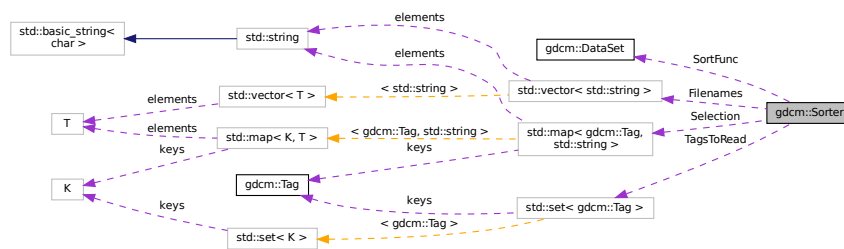
[Sorter.](#)

```
#include <gdcmSorter.h>
```

Inheritance diagram for gdcm::Sorter:



Collaboration diagram for gdcm::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)

### 10.279.1 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 everytime you call Sort, all files specified as input paramater are *read*

#### See also

[Scanner](#)

#### Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

## 10.279.2 Member Typedef Documentation

### 10.279.2.1 SelectionMap

```
typedef std::map<Tag, std::string> gdcM::Sorter::SelectionMap [protected]
```

### 10.279.2.2 SortFunction

```
typedef bool(* gdcM::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

## 10.279.3 Constructor & Destructor Documentation

### 10.279.3.1 Sorter()

```
gdcM::Sorter::Sorter ( )
```

### 10.279.3.2 ~Sorter()

```
virtual gdcM::Sorter::~~Sorter ( ) [virtual]
```

## 10.279.4 Member Function Documentation

### 10.279.4.1 AddSelect()

```
bool gdcM::Sorter::AddSelect (
    Tag const & tag,
    const char * value )
```

UNSUPPORTED FOR NOW.



#### 10.279.4.2 GetFileNames()

```
const std::vector<std::string>& gdcm::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](#), [gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.279.4.3 Print()

```
void gdcm::Sorter::Print (
    std::ostream & os ) const
```

Print.

##### Examples:

[gdcmorthoplanes.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

Referenced by `gdcm::operator<<()`.

#### 10.279.4.4 SetSortFunction()

```
void gdcm::Sorter::SetSortFunction (
    SortFunction f )
```

##### Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

#### 10.279.4.5 SetTagsToRead()

```
void gdcm::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.

#### 10.279.4.6 Sort()

```
virtual bool gdcm::Sorter::Sort (
    std::vector< std::string > const & filenames ) [virtual]
```

Typically the output of [Directory::GetFilenames\(\)](#)

Reimplemented in [gdcm::IPPSorter](#).

Examples:

[SortImage.cxx](#).

#### 10.279.4.7 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
    std::vector< std::string > const & filenames ) [virtual]
```

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 10.279.5 Friends And Related Function Documentation

#### 10.279.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Sorter & s ) [friend]
```

### 10.279.6 Member Data Documentation

#### 10.279.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames [protected]
```

### 10.279.6.2 Selection

```
std::map<Tag, std::string> gdcm::Sorter::Selection [protected]
```

### 10.279.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc [protected]
```

### 10.279.6.4 TagsToRead

```
std::set<Tag> gdcm::Sorter::TagsToRead [protected]
```

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

- [gdcmSorter.h](#)

## 10.280 gdcm::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcmSpacing.h>
```

### Public Types

- enum [SpacingType](#) {  
    [DETECTOR](#) = 0,  
    [MAGNIFIED](#),  
    [CALIBRATED](#),  
    [UNKNOWN](#) }

### Public Member Functions

- [Spacing](#) ()
- [~Spacing](#) ()

### Static Public Member Functions

- static [Attribute](#)< 0x28, 0x34 > [ComputePixelAspectRatioFromPixelSpacing](#) (const [Attribute](#)< 0x28, 0x30 > &pixelspacing)

### 10.280.1 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](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](http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing)

### 10.280.2 Member Enumeration Documentation

#### 10.280.2.1 SpacingType

enum [gdcm::Spacing::SpacingType](#)

## Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

### 10.280.3 Constructor & Destructor Documentation

#### 10.280.3.1 Spacing()

```
gdcm::Spacing::Spacing ( )
```

#### 10.280.3.2 ~Spacing()

```
gdcm::Spacing::~Spacing ( )
```

### 10.280.4 Member Function Documentation

#### 10.280.4.1 ComputePixelAspectRatioFromPixelSpacing()

```
static Attribute<0x28,0x34> gdcm::Spacing::ComputePixelAspectRatioFromPixelSpacing (
    const Attribute< 0x28, 0x30 > & pixelspacing ) [static]
```

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

- [gdcmSpacing.h](#)

## 10.281 gdcm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmSpectroscopy.h>
```

## Public Member Functions

- [Spectroscopy](#) ()

### 10.281.1 Detailed Description

[Spectroscopy](#) class.

### 10.281.2 Constructor & Destructor Documentation

#### 10.281.2.1 Spectroscopy()

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

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

- [gdcmSpectroscopy.h](#)

## 10.282 gdcm::SplitMosaicFilter Class Reference

[SplitMosaicFilter](#) class.

```
#include <gdcmSplitMosaicFilter.h>
```

## Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- bool [ComputeMOSAICSliceNormal](#) (double dims[3], bool &inverted)  
*Extract the value for SliceNormalVector (CSA header)*
- bool [ComputeMOSAICSlicePosition](#) (double pos[3], bool inverted)  
*Extract the value for ImagePositionPatient (requires inverted flag)*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- const [Image](#) & [GetImage](#) () const
- [Image](#) & [GetImage](#) ()
- void [SetFile](#) (const [File](#) &f)
- void [SetImage](#) (const [Image](#) &image)
- bool [Split](#) ()  
*Split the SIEMENS MOSAIC image.*

## 10.282.1 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

## 10.282.2 Constructor & Destructor Documentation

### 10.282.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ( )
```

### 10.282.2.2 ~SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::~~SplitMosaicFilter ( )
```

## 10.282.3 Member Function Documentation

### 10.282.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
    unsigned int dims[3] )
```

Compute the new dimensions according to private information stored in the MOSAIC header.

### 10.282.3.2 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (
    double dims[3],
    bool & inverted )
```

Extract the value for SliceNormalVector (CSA header)

### 10.282.3.3 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (
    double pos[3],
    bool inverted )
```

Extract the value for ImagePositionPatient (requires inverted flag)

### 10.282.3.4 GetFile() [1/2]

```
File& gdcm::SplitMosaicFilter::GetFile ( ) [inline]
```

### 10.282.3.5 GetFile() [2/2]

```
const File& gdcm::SplitMosaicFilter::GetFile ( ) const [inline]
```

### 10.282.3.6 GetImage() [1/2]

```
const Image& gdcm::SplitMosaicFilter::GetImage ( ) const [inline]
```

### 10.282.3.7 GetImage() [2/2]

```
Image& gdcm::SplitMosaicFilter::GetImage ( ) [inline]
```

### 10.282.3.8 SetFile()

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



### 10.282.3.9 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
    const Image & image )
```

### 10.282.3.10 Split()

```
bool gdcm::SplitMosaicFilter::Split ( )
```

Split the SIEMENS MOSAIC image.

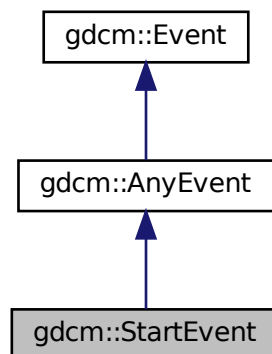
The documentation for this class was generated from the following file:

- [gdcmSplitMosaicFilter.h](#)

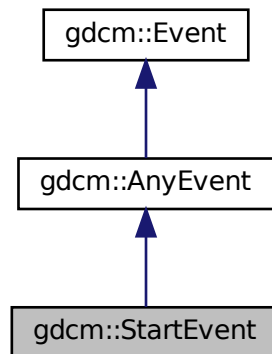
## 10.283 gdcm::StartEvent Class Reference

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

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for `gdcm::StartEvent`:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

### 10.284 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

### 10.285 `gdcm::STATIC_ASSERTION_FAILURE< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

## 10.286 gdcm::STATIC\_ASSERTION\_FAILURE< true > Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

### Public Types

- enum { [value](#) = 1 }

### 10.286.1 Member Enumeration Documentation

#### 10.286.1.1 anonymous enum

anonymous enum

#### Enumerator

value	
-------	--

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

- [gdcmStaticAssert.h](#)

## 10.287 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)

### 10.287.1 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:

[StreamImageReaderTest.cxx](#).

### 10.287.2 Constructor & Destructor Documentation

#### 10.287.2.1 StreamImageReader()

```
gdcm::StreamImageReader::StreamImageReader ( )
```

#### 10.287.2.2 ~StreamImageReader()

```
virtual gdcm::StreamImageReader::~~StreamImageReader ( ) [virtual]
```

### 10.287.3 Member Function Documentation

#### 10.287.3.1 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](#).

### 10.287.3.2 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:

[StreamImageReaderTest.cxx](#).

### 10.287.3.3 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:

[StreamImageReaderTest.cxx](#).

### 10.287.3.4 GetDimensionsValueForResolution()

```
std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (
    unsigned int )
```

### 10.287.3.5 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:

[StreamImageReaderTest.cxx](#).

### 10.287.3.6 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 `metainageio` in `itk` MUST have an extent defined, or else `Read` will return false. If no particular extent is required, use [ImageReader](#) instead.

Examples:

[StreamImageReaderTest.cxx](#).

### 10.287.3.7 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:

[StreamImageReaderTest.cxx](#).

### 10.287.3.8 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](#).

## 10.287.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (
    std::istream & inStream )
```

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

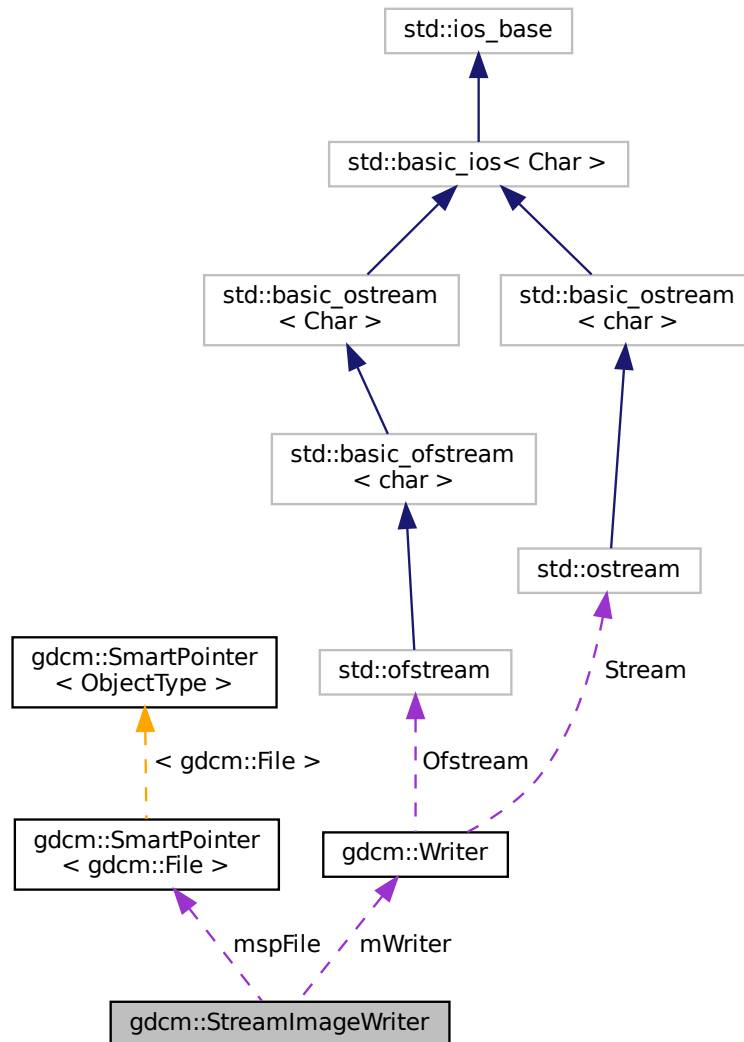
- [gdcmStreamImageReader.h](#)

## 10.288 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](#)

## 10.288.1 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](#).

## 10.288.2 Constructor & Destructor Documentation

### 10.288.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ( )
```

### 10.288.2.2 ~StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~~StreamImageWriter ( ) [virtual]
```

## 10.288.3 Member Function Documentation

### 10.288.3.1 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](#).

### 10.288.3.2 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](#).

### 10.288.3.3 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](#).

### 10.288.3.4 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](#).

### 10.288.3.5 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.

### 10.288.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (
    std::ostream & inStream )
```

#### Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.288.3.7 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 accomodate 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](#).

### 10.288.3.8 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](#).

### 10.288.3.9 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

### 10.288.3.10 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.

## 10.288.4 Member Data Documentation

### 10.288.4.1 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.

### 10.288.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1 [protected]
```

### 10.288.4.3 mspFile

```
SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]
```

### 10.288.4.4 mWriter

```
Writer gdcm::StreamImageWriter::mWriter [protected]
```

**10.288.4.5 mXMax**

uint16\_t gdcm::StreamImageWriter::mXMax [protected]

**10.288.4.6 mXMin**

uint16\_t gdcm::StreamImageWriter::mXMin [protected]

**10.288.4.7 mYMax**

uint16\_t gdcm::StreamImageWriter::mYMax [protected]

**10.288.4.8 mYMin**

uint16\_t gdcm::StreamImageWriter::mYMin [protected]

**10.288.4.9 mZMax**

uint16\_t gdcm::StreamImageWriter::mZMax [protected]

**10.288.4.10 mZMin**

uint16\_t gdcm::StreamImageWriter::mZMin [protected]

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

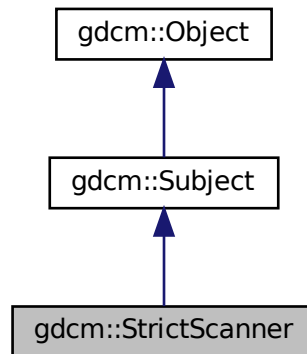
- [gdcmStreamImageWriter.h](#)

## 10.289 gdcm::StrictScanner Class Reference

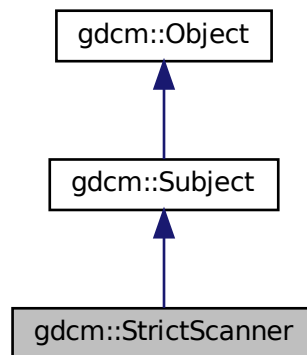
[StrictScanner](#).

```
#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](#), [Itstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char \* > [TagToValue](#)
- typedef TagToValue::value\_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

## Public Member Functions

- [StrictScanner](#) ()
- [~StrictScanner](#) ()
- 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  
*Print result.*
- bool [Scan](#) ([Directory::FileNamesType](#) const &filenames)  
*Start the scan !*

## 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)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [StrictScanner](#) &s)

### 10.289.1 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:

[SimpleScanner.cxx](#).

### 10.289.2 Member Typedef Documentation

#### 10.289.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

### 10.289.2.2 MappingType

```
typedef std::map<const char *, TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

### 10.289.2.3 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)

### 10.289.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

### 10.289.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

## 10.289.3 Constructor & Destructor Documentation

### 10.289.3.1 StrictScanner()

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

### 10.289.3.2 ~StrictScanner()

```
gdcm::StrictScanner::~~StrictScanner ( )
```

## 10.289.4 Member Function Documentation

#### 10.289.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (
    PrivateTag const & t )
```

#### 10.289.4.2 AddSkipTag()

```
void gdcm::StrictScanner::AddSkipTag (
    Tag const & t )
```

Add a tag that will need to be skipped. Those are root level skip tags.

#### 10.289.4.3 AddTag()

```
void gdcm::StrictScanner::AddTag (
    Tag const & t )
```

Add a tag that will need to be read. Those are root level skip tags.

Examples:

[SimpleScanner.cxx](#).

#### 10.289.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin ( ) const [inline]
```

#### 10.289.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ( )
```

#### 10.289.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ( )
```

**10.289.4.7 End()**

```
ConstIterator gdcM::StrictScanner::End ( ) const [inline]
```

**10.289.4.8 GetAllFileNamesFromTagToValue()**

```
Directory::FileNamesType gdcM::StrictScanner::GetAllFileNamesFromTagToValue (
    Tag const & t,
    const char * valueref ) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

**10.289.4.9 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'

**10.289.4.10 GetFileNames()**

```
Directory::FileNamesType const& gdcM::StrictScanner::GetFileNames ( ) const [inline]
```

**10.289.4.11 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

**10.289.4.12 GetMapping()**

```
TagToValue const& gdcM::StrictScanner::GetMapping (
    const char * filename ) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples:

[SimpleScanner.cxx](#).

#### 10.289.4.13 GetMappingFromTagToValue()

```
TagToValue const& gdcm::StrictScanner::GetMappingFromTagToValue (
    Tag const & t,
    const char * value ) const
```

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

#### 10.289.4.14 GetMappings()

```
MappingType const& gdcm::StrictScanner::GetMappings ( ) const [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

#### 10.289.4.15 GetOrderedValues()

```
Directory::FileNamesType gdcm::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)

#### 10.289.4.16 GetValue()

```
const char* gdcm::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 !

#### 10.289.4.17 GetValues() [1/2]

```
ValueType const& gdcm::StrictScanner::GetValues ( ) const [inline]
```

Get all the values found (in lexicographic order)

**10.289.4.18** **GetValues()** [2/2]

```
ValueType gdcM::StrictScanner::GetValues (
    Tag const & t ) const
```

Get all the values found (in lexicographic order) associated with [Tag](#) 't'.

**10.289.4.19** **IsKey()**

```
bool gdcM::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:

[SimpleScanner.cxx](#).

**10.289.4.20** **New()**

```
static SmartPointer<StrictScanner> gdcM::StrictScanner::New ( ) [inline], [static]
```

for wrapped language: instantiate a reference counted object

**10.289.4.21** **Print()**

```
void gdcM::StrictScanner::Print (
    std::ostream & os ) const [virtual]
```

Print result.

Reimplemented from [gdcM::Object](#).

Referenced by [gdcM::operator<<\(\)](#).

#### 10.289.4.22 ProcessPublicTag()

```
void gdcm::StrictScanner::ProcessPublicTag (
    StringFilter & sf,
    const char * filename ) [protected]
```

#### 10.289.4.23 Scan()

```
bool gdcm::StrictScanner::Scan (
    Directory::FilenameType const & filenames )
```

Start the scan !

Examples:

[SimpleScanner.cxx](#).

### 10.289.5 Friends And Related Function Documentation

#### 10.289.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const StrictScanner & s ) [friend]
```

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

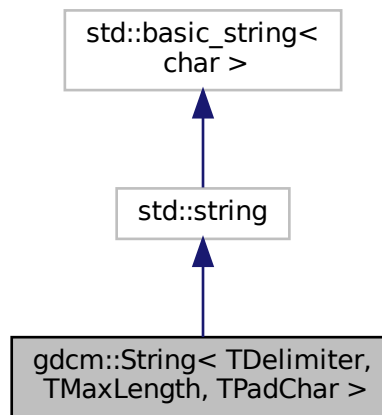
- [gdcmStrictScanner.h](#)

## 10.290 `gdcm::String< TDelimiter, TMaxLength, TPadChar >` Class Template Reference

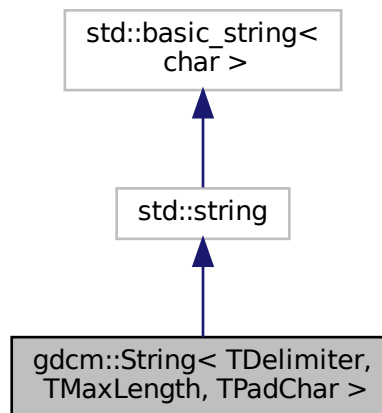
[String.](#)

```
#include <gdcmString.h>
```

Inheritance diagram for `gdcm::String< TDelimiter, TMaxLength, TPadChar >`:



Collaboration diagram for `gdcm::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 [value\\_type](#) \*s)
- [String](#) (const [value\\_type](#) \*s, [size\\_type](#) n)
- [String](#) (const std::string &s, [size\\_type](#) pos=0, [size\\_type](#) n=npos)
- 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
- [gdcm::String](#)< TDelimiter, TMaxLength, TPadChar > [Truncate](#) () const

## Static Public Member Functions

- static std::string [Trim](#) (const char \*input)

### 10.290.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

#### Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

#### Examples:

[TemplateEmptyImage.cxx](#).

## 10.290.2 Member Typedef Documentation

### 10.290.2.1 const\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↵
iterator
```

### 10.290.2.2 const\_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↵
reference
```

### 10.290.2.3 const\_reverse\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >↵
::const_reverse_iterator
```

### 10.290.2.4 difference\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference↵
_type
```

### 10.290.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

### 10.290.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

### 10.290.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

### 10.290.2.8 reverse\_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse↵
_iterator
```

### 10.290.2.9 size\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

### 10.290.2.10 value\_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

## 10.290.3 Constructor & Destructor Documentation

### 10.290.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]
```

String constructors.

**10.290.3.2 String()** [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value_type * s ) [inline]
```

**10.290.3.3 String()** [3/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const value_type * s,
    size_type n ) [inline]
```

**10.290.3.4 String()** [4/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcmm::String< TDelimiter, TMaxLength, TPadChar >::String (
    const std::string & s,
    size_type pos = 0,
    size_type n = npos ) [inline]
```

**10.290.4 Member Function Documentation****10.290.4.1 IsValid()**

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
bool gdcmm::String< TDelimiter, TMaxLength, TPadChar >::IsValid ( ) const [inline]
```

return if string is valid

Referenced by gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Truncate().

**10.290.4.2 operator const char \*()**

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcmm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * ( ) const [inline]
```

WARNING: Trailing \0 might be lost in this operation:

## 10.290.4.3 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...

## 10.290.4.4 Trim() [2/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
static std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (
    const char * input ) [inline], [static]
```

## 10.290.4.5 Truncate()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String<TDelimiter, TMaxLength, TPadChar> gdcm::String< TDelimiter, TMaxLength, TPadChar
>::Truncate ( ) const [inline]
```

References [gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid\(\)](#).

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

- [gdcmString.h](#)

## 10.291 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 [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

### 10.291.1 Detailed Description

[StringFilter](#).

[StringFilter](#) 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

Examples:

[ReadAndPrintAttributes.cxx](#).

### 10.291.2 Constructor & Destructor Documentation

#### 10.291.2.1 [StringFilter\(\)](#)

```
gdcm::StringFilter::StringFilter ( )
```

#### 10.291.2.2 [~StringFilter\(\)](#)

```
gdcm::StringFilter::~~StringFilter ( )
```

### 10.291.3 Member Function Documentation

#### 10.291.3.1 [ExecuteQuery\(\)](#) [1/2]

```
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 syntatically correct

### 10.291.3.2 ExecuteQuery() [2/2]

```
bool gdcmm::StringFilter::ExecuteQuery (
    std::string const & query,
    DataSet const & ds,
    std::string & value ) const [protected]
```

### 10.291.3.3 FromString()

```
std::string gdcmm::StringFilter::FromString (
    const Tag & t,
    const char * value,
    size_t len )
```

Convert to string the char array defined by the pair (value,len)

### 10.291.3.4 GetFile() [1/2]

```
File& gdcmm::StringFilter::GetFile ( ) [inline]
```

### 10.291.3.5 GetFile() [2/2]

```
const File& gdcmm::StringFilter::GetFile ( ) const [inline]
```

### 10.291.3.6 SetDicts()

```
void gdcmm::StringFilter::SetDicts (
    const Dicts & dicts )
```

Allow user to pass in there own dicts.

### 10.291.3.7 SetFile()

```
void gdcM::StringFilter::SetFile (
    const File & f ) [inline]
```

Set/Get [File](#).

Examples:

[ReadAndPrintAttributes.cxx](#), and [SimplePrintPatientName.cs](#).

### 10.291.3.8 ToString() [1/2]

```
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:

[ReadAndPrintAttributes.cxx](#).

### 10.291.3.9 ToString() [2/2]

```
std::string gdcM::StringFilter::ToString (
    const Tag & t ) const
```

Directly from a [Tag](#):

### 10.291.3.10 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](#) pair.second : the value encoded into a string (US,UL...) are properly converted

Examples:

[ReadAndPrintAttributes.cxx](#).



**10.291.3.11 ToStringPair()** [2/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
    const Tag & t ) const
```

Directly from a [Tag](#):

**10.291.3.12 ToStringPair()** [3/3]

```
std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (
    const Tag & t,
    DataSet const & ds ) const [protected]
```

**10.291.3.13 UseDictAlways()**

```
void gdcm::StringFilter::UseDictAlways (
    bool ) [inline]
```

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

- [gdcmStringFilter.h](#)

## 10.292 gdcm::Study Class Reference

[Study](#).

```
#include <gdcmStudy.h>
```

### Public Member Functions

- [Study](#) ()

### 10.292.1 Detailed Description

[Study](#).

### 10.292.2 Constructor & Destructor Documentation

### 10.292.2.1 Study()

```
gdcM::Study::Study ( ) [inline]
```

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

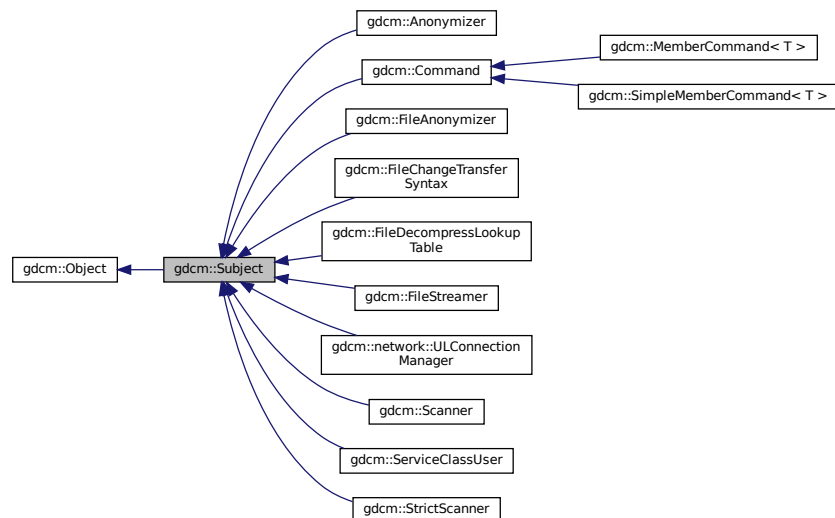
- [gdcMStudy.h](#)

## 10.293 gdcM::Subject Class Reference

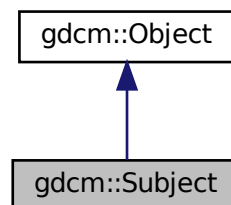
[Subject.](#)

```
#include <gdcMSubject.h>
```

Inheritance diagram for gdcM::Subject:



Collaboration diagram for gdcM::Subject:



## Public Member Functions

- [Subject](#) ()
- [~Subject](#) ()
- 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)

## Additional Inherited Members

### 10.293.1 Detailed Description

[Subject](#).

See also

[Command Event](#)

Examples:

[SimpleScanner.cxx](#).

### 10.293.2 Constructor & Destructor Documentation

#### 10.293.2.1 Subject()

```
gdcmm::Subject::Subject ( )
```

#### 10.293.2.2 ~Subject()

```
gdcmm::Subject::~~Subject ( )
```

### 10.293.3 Member Function Documentation

**10.293.3.1 AddObserver()** [1/2]

```
unsigned long gdcM::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 [gdcM::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

**10.293.3.2 AddObserver()** [2/2]

```
unsigned long gdcM::Subject::AddObserver (
    const Event & event,
    Command * ) const
```

**10.293.3.3 GetCommand()**

```
Command* gdcM::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.

**10.293.3.4 HasObserver()**

```
bool gdcM::Subject::HasObserver (
    const Event & event ) const
```

Return true if an observer is registered for this event.

**10.293.3.5 InvokeEvent()** [1/2]

```
void gdcM::Subject::InvokeEvent (
    const Event & )
```

Call Execute on all the Commands observing this event id.

### 10.293.3.6 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.

### 10.293.3.7 RemoveAllObservers()

```
void gdcm::Subject::RemoveAllObservers ( )
```

Remove all observers .

### 10.293.3.8 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:

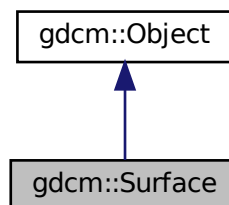
- [gdcmSubject.h](#)

## 10.294 gdcm::Surface Class Reference

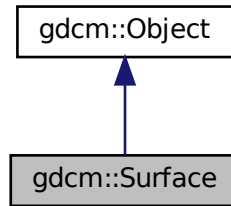
This class defines a SURFACE IE.

```
#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 ()`
- virtual `~Surface ()`
- `SegmentHelper::BasicCodedEntry` const & `GetAlgorithmFamily ()` const
- `SegmentHelper::BasicCodedEntry` & `GetAlgorithmFamily ()`
- 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` const & `GetMeshPrimitive ()` const
- `MeshPrimitive` & `GetMeshPrimitive ()`
- unsigned long `GetNumberOfSurfacePoints ()` const
- unsigned long `GetNumberOfVectors ()` const
- const `DataElement` & `GetPointCoordinatesData ()` const

- [DataElement](#) & [GetPointCoordinatesData](#) ()
- const float \* [GetPointPositionAccuracy](#) () const
- const float \* [GetPointsBoundingBoxCoordinates](#) () const
- [SegmentHelper::BasicCodedEntry](#) const & [GetProcessingAlgorithm](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetProcessingAlgorithm](#) ()
- 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
- const [DataElement](#) & [GetVectorCoordinateData](#) () const
- [DataElement](#) & [GetVectorCoordinateData](#) ()
- 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](#) &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 unsigned short vl[3])
- void [SetRecommendedDisplayCIELabValue](#) (const unsigned short vl, const unsigned int idx=0)
- void [SetRecommendedDisplayCIELabValue](#) (const std::vector< unsigned short > &vl)
- 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)

## 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

### 10.294.1 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

### 10.294.2 Member Enumeration Documentation

#### 10.294.2.1 STATES

```
enum gdcm::Surface::STATES
```

Enumerator

NO	
YES	
UNKNOWN	
STATES_END	

#### 10.294.2.2 VIEWType

```
enum gdcm::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	

### 10.294.3 Constructor & Destructor Documentation

#### 10.294.3.1 Surface()

```
gdcm::Surface::Surface ( )
```

#### 10.294.3.2 ~Surface()

```
virtual gdcm::Surface::~~Surface ( ) [virtual]
```

### 10.294.4 Member Function Documentation

#### 10.294.4.1 GetAlgorithmFamily() [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const
```

#### 10.294.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )
```

#### 10.294.4.3 GetAlgorithmName()

```
const char* gdcm::Surface::GetAlgorithmName ( ) const
```

**10.294.4.4 GetAlgorithmVersion()**

```
const char* gdcm::Surface::GetAlgorithmVersion ( ) const
```

**10.294.4.5 GetAxisOfRotation()**

```
const float* gdcm::Surface::GetAxisOfRotation ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.6 GetCenterOfRotation()**

```
const float* gdcm::Surface::GetCenterOfRotation ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.7 GetFiniteVolume()**

```
STATES gdcm::Surface::GetFiniteVolume ( ) const
```

**10.294.4.8 GetManifold()**

```
STATES gdcm::Surface::GetManifold ( ) const
```

**10.294.4.9 GetMaximumPointDistance()**

```
float gdcm::Surface::GetMaximumPointDistance ( ) const
```

**10.294.4.10 GetMeanPointDistance()**

```
float gdcm::Surface::GetMeanPointDistance ( ) const
```

**10.294.4.11 GetMeshPrimitive()** [1/2]

```
MeshPrimitive const& gdcm::Surface::GetMeshPrimitive ( ) const
```

**10.294.4.12 GetMeshPrimitive()** [2/2]

```
MeshPrimitive& gdcm::Surface::GetMeshPrimitive ( )
```

**10.294.4.13 GetNumberOfSurfacePoints()**

```
unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const
```

**10.294.4.14 GetNumberOfVectors()**

```
unsigned long gdcm::Surface::GetNumberOfVectors ( ) const
```

**10.294.4.15 GetPointCoordinatesData()** [1/2]

```
const DataElement& gdcm::Surface::GetPointCoordinatesData ( ) const
```

**10.294.4.16 GetPointCoordinatesData()** [2/2]

```
DataElement& gdcm::Surface::GetPointCoordinatesData ( )
```

**10.294.4.17 GetPointPositionAccuracy()**

```
const float* gdcm::Surface::GetPointPositionAccuracy ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.18 GetPointsBoundingBoxCoordinates()**

```
const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const
```

**Note**

Pointer is null if undefined

**10.294.4.19 GetProcessingAlgorithm()** [1/2]

```
SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetProcessingAlgorithm ( ) const
```

**10.294.4.20 GetProcessingAlgorithm()** [2/2]

```
SegmentHelper::BasicCodedEntry& gdcm::Surface::GetProcessingAlgorithm ( )
```

**10.294.4.21 GetRecommendedDisplayCIELabValue()** [1/2]

```
const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const
```

**10.294.4.22 GetRecommendedDisplayCIELabValue()** [2/2]

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
    const unsigned int idx ) const
```

**10.294.4.23 GetRecommendedDisplayGrayscaleValue()**

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
```

**10.294.4.24 GetRecommendedPresentationOpacity()**

```
float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const
```

**10.294.4.25 GetRecommendedPresentationType()**

```
VIEWType gdcm::Surface::GetRecommendedPresentationType ( ) const
```

**10.294.4.26 GetSTATES()**

```
static STATES gdcm::Surface::GetSTATES (
    const char * state ) [static]
```

**10.294.4.27 GetSTATESString()**

```
static const char* gdcm::Surface::GetSTATESString (
    STATES state ) [static]
```

**10.294.4.28 GetSurfaceComments()**

```
const char* gdcm::Surface::GetSurfaceComments ( ) const
```

**10.294.4.29 GetSurfaceNumber()**

```
unsigned long gdcm::Surface::GetSurfaceNumber ( ) const
```

**10.294.4.30 GetSurfaceProcessing()**

```
bool gdcm::Surface::GetSurfaceProcessing ( ) const
```

**10.294.4.31 GetSurfaceProcessingDescription()**

```
const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const
```

**10.294.4.32 GetSurfaceProcessingRatio()**

```
float gdcm::Surface::GetSurfaceProcessingRatio ( ) const
```

**10.294.4.33 GetVectorAccuracy()**

```
const float* gdcm::Surface::GetVectorAccuracy ( ) const
```

**10.294.4.34 GetVectorCoordinateData()** [1/2]

```
const DataElement& gdcm::Surface::GetVectorCoordinateData ( ) const
```

**10.294.4.35 GetVectorCoordinateData()** [2/2]

```
DataElement& gdcm::Surface::GetVectorCoordinateData ( )
```

**10.294.4.36 GetVectorDimensionality()**

```
unsigned short gdcm::Surface::GetVectorDimensionality ( ) const
```

#### 10.294.4.37 GetVIEWType()

```
static VIEWType gdcm::Surface::GetVIEWType (
    const char * type ) [static]
```

#### 10.294.4.38 GetVIEWTypeString()

```
static const char* gdcm::Surface::GetVIEWTypeString (
    VIEWType type ) [static]
```

#### 10.294.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
    SegmentHelper::BasicCodedEntry const & BSE )
```

#### 10.294.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
    const char * str )
```

#### 10.294.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
    const char * str )
```

#### 10.294.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
    const float * axis )
```

**10.294.4.43 SetCenterOfRotation()**

```
void gdcM::Surface::SetCenterOfRotation (
    const float * center )
```

**10.294.4.44 SetFiniteVolume()**

```
void gdcM::Surface::SetFiniteVolume (
    STATES state )
```

**10.294.4.45 SetManifold()**

```
void gdcM::Surface::SetManifold (
    STATES state )
```

**10.294.4.46 SetMaximumPointDistance()**

```
void gdcM::Surface::SetMaximumPointDistance (
    float maximum )
```

**10.294.4.47 SetMeanPointDistance()**

```
void gdcM::Surface::SetMeanPointDistance (
    float average )
```

**10.294.4.48 SetMeshPrimitive()**

```
void gdcM::Surface::SetMeshPrimitive (
    MeshPrimitive & mp )
```



**10.294.4.49 SetNumberOfSurfacePoints()**

```
void gdcm::Surface::SetNumberOfSurfacePoints (
    const unsigned long nb )
```

**10.294.4.50 SetNumberOfVectors()**

```
void gdcm::Surface::SetNumberOfVectors (
    const unsigned long nb )
```

**10.294.4.51 SetPointCoordinatesData()**

```
void gdcm::Surface::SetPointCoordinatesData (
    DataElement const & de )
```

**10.294.4.52 SetPointPositionAccuracy()**

```
void gdcm::Surface::SetPointPositionAccuracy (
    const float * accuracies )
```

**10.294.4.53 SetPointsBoundingBoxCoordinates()**

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
    const float * coordinates )
```

**10.294.4.54 SetProcessingAlgorithm()**

```
void gdcm::Surface::SetProcessingAlgorithm (
    SegmentHelper::BasicCodedEntry const & BSE )
```

**10.294.4.55 SetRecommendedDisplayCIELabValue()** [1/3]

```
void gdcM::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl[3] )
```

**10.294.4.56 SetRecommendedDisplayCIELabValue()** [2/3]

```
void gdcM::Surface::SetRecommendedDisplayCIELabValue (
    const unsigned short vl,
    const unsigned int idx = 0 )
```

**10.294.4.57 SetRecommendedDisplayCIELabValue()** [3/3]

```
void gdcM::Surface::SetRecommendedDisplayCIELabValue (
    const std::vector< unsigned short > & vl )
```

**10.294.4.58 SetRecommendedDisplayGrayscaleValue()**

```
void gdcM::Surface::SetRecommendedDisplayGrayscaleValue (
    const unsigned short vl )
```

**10.294.4.59 SetRecommendedPresentationOpacity()**

```
void gdcM::Surface::SetRecommendedPresentationOpacity (
    const float opacity )
```

**10.294.4.60 SetRecommendedPresentationType()**

```
void gdcM::Surface::SetRecommendedPresentationType (
    VIEWType type )
```

**10.294.4.61 SetSurfaceComments()**

```
void gdcm::Surface::SetSurfaceComments (
    const char * comment )
```

**10.294.4.62 SetSurfaceNumber()**

```
void gdcm::Surface::SetSurfaceNumber (
    const unsigned long nb )
```

**10.294.4.63 SetSurfaceProcessing()**

```
void gdcm::Surface::SetSurfaceProcessing (
    bool b )
```

**10.294.4.64 SetSurfaceProcessingDescription()**

```
void gdcm::Surface::SetSurfaceProcessingDescription (
    const char * description )
```

**10.294.4.65 SetSurfaceProcessingRatio()**

```
void gdcm::Surface::SetSurfaceProcessingRatio (
    const float ratio )
```

**10.294.4.66 SetVectorAccuracy()**

```
void gdcm::Surface::SetVectorAccuracy (
    const float * accuracy )
```

**10.294.4.67 SetVectorCoordinateData()**

```
void gdcmm::Surface::SetVectorCoordinateData (
    DataElement const & de )
```

**10.294.4.68 SetVectorDimensionality()**

```
void gdcmm::Surface::SetVectorDimensionality (
    const unsigned short dim )
```

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

- [gdcmmSurface.h](#)

**10.295 gdcmm::SurfaceHelper Class Reference**

[SurfaceHelper](#).

```
#include <gdcmmSurfaceHelper.h>
```

**Public Types**

- typedef std::vector< unsigned short > [ColorArray](#)

**Static Public Member Functions**

- template<typename T, typename U >  
static std::vector< T > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range←  
Max=255)  
*Convert a DICOM CIE-Lab (after reading) color into RGB.*
- template<typename U >  
static std::vector< float > [RecommendedDisplayCIELabToRGB](#) (const [ColorArray](#) &CIELab, const U range←  
Max=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 range←  
Max=255)  
*Convert a RGB color into DICOM grayscale (ready to write).*

### 10.295.1 Detailed Description

[SurfaceHelper](#).

Helper class for [Surface](#) object

### 10.295.2 Member Typedef Documentation

#### 10.295.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

### 10.295.3 Member Function Documentation

#### 10.295.3.1 RecommendedDisplayCIELabToRGB() [1/2]

```
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

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

#### Template Parameters

<i>T</i>	<a href="#">Type</a> of CIELab components.
<i>U</i>	<a href="#">Type</a> of rangeMax value.

**10.295.3.2 RecommendedDisplayCIELabToRGB()** [2/2]

```
template<typename U >
std::vector< float > gdc::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**

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

**Template Parameters**

<i>U</i>	<a href="#">Type</a> of rangeMax value.
----------	---

**10.295.3.3 RGBToRecommendedDisplayCIELab()**

```
template<typename T , typename U >
SurfaceHelper::ColorArray gdc::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**

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

**Template Parameters**

<i>T</i>	<a href="#">Type</a> of RGB components.
----------	---

## Template Parameters

<i>U</i>	Type of rangeMax value.
----------	-------------------------

## 10.295.3.4 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

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

## Template Parameters

<i>T</i>	Type of RGB components.
<i>U</i>	Type of rangeMax value.

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

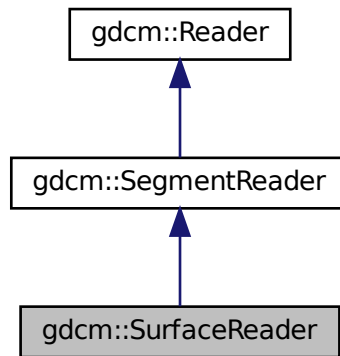
- [gdcmSurfaceHelper.h](#)

## 10.296 gdcm::SurfaceReader Class Reference

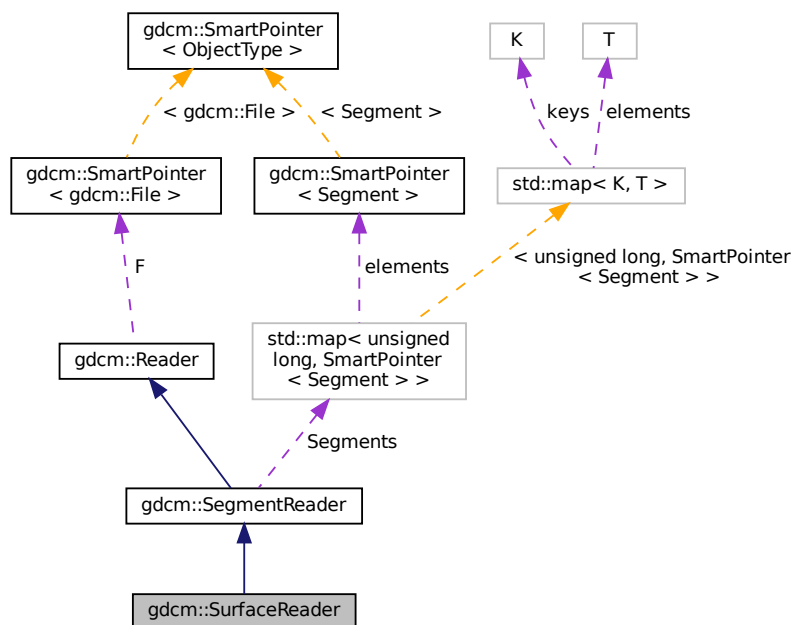
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for `gdcM::SurfaceReader`:



Collaboration diagram for `gdcM::SurfaceReader`:



## Public Member Functions

- [SurfaceReader](#) ()



- virtual [~SurfaceReader](#) ()
- unsigned long [GetNumberOfSurfaces](#) () const
- virtual bool [Read](#) ()  
*Read.*

### Protected Member Functions

- bool [ReadPointMacro](#) ([SmartPointer](#)< [Surface](#) > surface, const [DataSet](#) &surfaceDS)
- bool [ReadSurface](#) (const [Item](#) &surfaceItem, const unsigned long idx)
- bool [ReadSurfaces](#) ()

### Additional Inherited Members

#### 10.296.1 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

#### 10.296.2 Constructor & Destructor Documentation

##### 10.296.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ( )
```

##### 10.296.2.2 ~SurfaceReader()

```
virtual gdcm::SurfaceReader::~~SurfaceReader ( ) [virtual]
```

#### 10.296.3 Member Function Documentation

### 10.296.3.1 GetNumberOfSurfaces()

```
unsigned long gdcM::SurfaceReader::GetNumberOfSurfaces ( ) const
```

### 10.296.3.2 Read()

```
virtual bool gdcM::SurfaceReader::Read ( ) [virtual]
```

Read.

Reimplemented from [gdcM::SegmentReader](#).

### 10.296.3.3 ReadPointMacro()

```
bool gdcM::SurfaceReader::ReadPointMacro (
    SmartPointer< Surface > surface,
    const DataSet & surfaceDS ) [protected]
```

### 10.296.3.4 ReadSurface()

```
bool gdcM::SurfaceReader::ReadSurface (
    const Item & surfaceItem,
    const unsigned long idx ) [protected]
```

### 10.296.3.5 ReadSurfaces()

```
bool gdcM::SurfaceReader::ReadSurfaces ( ) [protected]
```

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

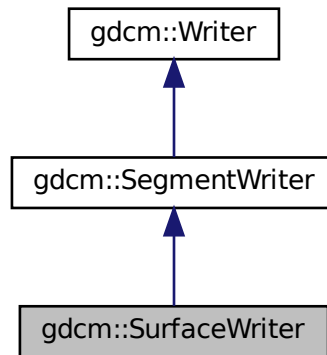
- [gdcMSurfaceReader.h](#)

## 10.297 gdcm::SurfaceWriter Class Reference

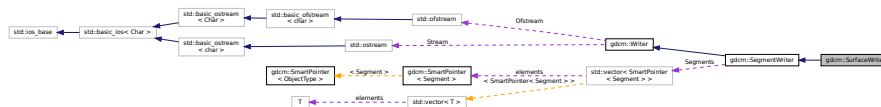
This class defines a SURFACE IE writer.

```
#include <gdcmSurfaceWriter.h>
```

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



### Public Member Functions

- [SurfaceWriter](#) ()
- virtual [~SurfaceWriter](#) ()
- unsigned long [GetNumberOfSurfaces](#) ()
- void [SetNumberOfSurfaces](#) (const unsigned long nb)
- bool [Write](#) ()

*Write.*

### Protected Member Functions

- void [ComputeNumberOfSurfaces](#) ()
- bool [PrepareWrite](#) ()
- bool [PrepareWritePointMacro](#) (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)

## Protected Attributes

- unsigned long [NumberOfSurfaces](#)

## Additional Inherited Members

### 10.297.1 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

### 10.297.2 Constructor & Destructor Documentation

#### 10.297.2.1 SurfaceWriter()

```
gdcM::SurfaceWriter::SurfaceWriter ( )
```

#### 10.297.2.2 ~SurfaceWriter()

```
virtual gdcM::SurfaceWriter::~~SurfaceWriter ( ) [virtual]
```

### 10.297.3 Member Function Documentation

#### 10.297.3.1 ComputeNumberOfSurfaces()

```
void gdcM::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]
```

### 10.297.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )
```

### 10.297.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]
```

### 10.297.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
    SmartPointer< Surface > surface,
    DataSet & surfaceDS,
    const TransferSyntax & ts ) [protected]
```

### 10.297.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
    const unsigned long nb )
```

### 10.297.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ( ) [virtual]
```

Write.

Reimplemented from [gdcm::SegmentWriter](#).

## 10.297.4 Member Data Documentation

#### 10.297.4.1 NumberOfSurfaces

```
unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]
```

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

- [gdcmSurfaceWriter.h](#)

### 10.298 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)

## 10.298.1 Detailed Description

[SwapCode](#) representation.

Examples:

[TestByteSwap.cxx](#).

## 10.298.2 Member Enumeration Documentation

### 10.298.2.1 SwapCodeType

```
enum gdcm::SwapCode::SwapCodeType
```

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

## 10.298.3 Constructor & Destructor Documentation

### 10.298.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (
    SwapCodeType sc = Unknown ) [inline]
```

References [gdcm::operator<<\(\)](#).

## 10.298.4 Member Function Documentation

#### 10.298.4.1 GetIndex()

```
static int gdcM::SwapCode::GetIndex (
    SwapCode const & sc ) [static], [protected]
```

#### 10.298.4.2 GetSwapCodeString()

```
static const char* gdcM::SwapCode::GetSwapCodeString (
    SwapCode const & sc ) [static]
```

Referenced by gdcM::operator<<().

#### 10.298.4.3 operator SwapCode::SwapCodeType()

```
gdcM::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]
```

### 10.298.5 Friends And Related Function Documentation

#### 10.298.5.1 operator<<

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

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

- [gdcMSwapCode.h](#)

## 10.299 gdcM::SwapperDoOp 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)



## 10.299.1 Member Function Documentation

### 10.299.1.1 Swap()

```
template<typename T >
static T gdcm::SwapperDoOp::Swap (
    T val ) [static]
```

### 10.299.1.2 SwapArray()

```
template<typename T >
static void gdcm::SwapperDoOp::SwapArray (
    T * array,
    size_t n ) [inline], [static]
```

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

- [gdcmSwapper.h](#)

## 10.300 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)

### 10.300.1 Detailed Description

Examples:

[DumpSiemensBase64.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

### 10.300.2 Member Function Documentation

### 10.300.2.1 Swap()

```
template<typename T >
static T gdcm::SwapperNoOp::Swap (
    T val ) [inline], [static]
```

### 10.300.2.2 SwapArray()

```
template<typename T >
static void gdcm::SwapperNoOp::SwapArray (
    T * ,
    size_t ) [inline], [static]
```

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

- [gdcmSwapper.h](#)

## 10.301 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

### Static Public Member Functions

- 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 sytem.*
- 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)

### 10.301.1 Detailed Description

Class to do system operation.

OS independent functionalities

### 10.301.2 Member Function Documentation

#### 10.301.2.1 DeleteDirectory()

```
static bool gdcmm::System::DeleteDirectory (
    const char * source ) [static]
```

remove a directory named source

### 10.301.2.2 EncodeBytes()

```
static 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

### 10.301.2.3 FileExists()

```
static bool gdcM::System::FileExists (
    const char * filename ) [static]
```

Check whether the specified file exist on the sytem.

#### Examples:

[EncapsulateFileInRawData.cxx](#), [gdcMorthoplanes.cxx](#), and [MagnifyFile.cxx](#).

### 10.301.2.4 FileIsDirectory()

```
static bool gdcM::System::FileIsDirectory (
    const char * name ) [static]
```

Check whether the file specified is a directory:

#### Examples:

[gdcMorthoplanes.cxx](#), and [threadgdcM.cxx](#).

### 10.301.2.5 FileIsSymlink()

```
static bool gdcM::System::FileIsSymlink (
    const char * name ) [static]
```

Check whether name is a symlink.

### 10.301.2.6 FileSize()

```
static size_t gdcm::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](#).

### 10.301.2.7 FileTime()

```
static time_t gdcm::System::FileTime (
    const char * filename ) [static]
```

Return the time of last modification of file 0 if the file does not exist

### 10.301.2.8 FormatDateTime()

```
static 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]

### 10.301.2.9 GetCurrentDateTime()

```
static 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](#).

**10.301.2.10 GetCurrentModuleFileName()**

```
static const char* gdcM::System::GetCurrentModuleFileName ( ) [static]
```

Return the directory the current module is located: NOT THREAD SAFE

**10.301.2.11 GetCurrentProcessFileName()**

```
static const char* gdcM::System::GetCurrentProcessFileName ( ) [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

**10.301.2.12 GetCurrentResourcesDirectory()**

```
static const char* gdcM::System::GetCurrentResourcesDirectory ( ) [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

**10.301.2.13 GetCWD()**

```
static const char* gdcM::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

**10.301.2.14 GetHostName()**

```
static bool gdcM::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

**10.301.2.15 GetLastSystemError()**

```
static const char* gdcM::System::GetLastSystemError ( ) [static]
```

Return the last error.

#### 10.301.2.16 GetLocaleCharset()

```
static const char* gdcmm::System::GetLocaleCharset ( ) [static]
```

return locale charmap

#### 10.301.2.17 GetPermissions()

```
static bool gdcmm::System::GetPermissions (
    const char * file,
    unsigned short & mode ) [static], [protected]
```

NOT THREAD SAFE.

#### 10.301.2.18 GetTimezoneOffsetFromUTC()

```
static const char* gdcmm::System::GetTimezoneOffsetFromUTC ( ) [static]
```

Return the value for Timezone Offset From UTC as string.

##### Warning

not thread safe

#### 10.301.2.19 MakeDirectory()

```
static bool gdcmm::System::MakeDirectory (
    const char * path ) [static]
```

Create a directory name path.

#### 10.301.2.20 ParseDateTime() [1/2]

```
static bool gdcmm::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)

**10.301.2.21 ParseDateTime()** [2/2]

```
static 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](#)

**10.301.2.22 RemoveFile()**

```
static bool gdcm::System::RemoveFile (
    const char * source ) [static]
```

remove a file named source

**10.301.2.23 SetPermissions()**

```
static bool gdcm::System::SetPermissions (
    const char * file,
    unsigned short mode ) [static], [protected]
```

**10.301.2.24 StrCaseCmp()**

```
static int gdcm::System::StrCaseCmp (
    const char * s1,
    const char * s2 ) [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp



### 10.301.2.25 StrNCaseCmp()

```
static int gdcm::System::StrNCaseCmp (
    const char * s1,
    const char * s2,
    size_t n ) [static]
```

#### Precondition

`n != 0`

### 10.301.2.26 StrSep()

```
static 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

### 10.301.2.27 StrTokR()

```
static 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](#)

## 10.302 gdcm::Table Class Reference

[Table](#).

```
#include <gdcmTable.h>
```

#### Public Types

- typedef std::map< [Tag](#), [TableEntry](#) > [MapTableEntry](#)

## Public Member Functions

- [Table](#) ()
- [~Table](#) ()
- const [TableEntry](#) & [GetTableEntry](#) (const [Tag](#) &tag) const
- void [InsertEntry](#) ([Tag](#) const &tag, [TableEntry](#) const &te)

## Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Table](#) &\_val)

### 10.302.1 Detailed Description

[Table](#).

### 10.302.2 Member Typedef Documentation

#### 10.302.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcmm::Table::MapTableEntry
```

### 10.302.3 Constructor & Destructor Documentation

#### 10.302.3.1 Table()

```
gdcmm::Table::Table ( ) [inline]
```

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

#### 10.302.3.2 ~Table()

```
gdcmm::Table::~~Table ( ) [inline]
```

References [operator<<](#).

## 10.302.4 Member Function Documentation

### 10.302.4.1 GetTableEntry()

```
const TableEntry& gdcm::Table::GetTableEntry (  
    const Tag & tag ) const [inline]
```

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

### 10.302.4.2 InsertEntry()

```
void gdcm::Table::InsertEntry (  
    Tag const & tag,  
    TableEntry const & te ) [inline]
```

## 10.302.5 Friends And Related Function Documentation

### 10.302.5.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & _os,  
    const Table & _val ) [friend]
```

Referenced by [~Table\(\)](#).

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

- [gdcmTable.h](#)

## 10.303 gdcm::TableEntry Class Reference

[TableEntry](#).

```
#include <gdcmTableEntry.h>
```

## Public Member Functions

- [TableEntry](#) (const char \*attribute=0, [Type](#) const &type=[Type](#)(), const char \*des=0)
- [~TableEntry](#) ()

### 10.303.1 Detailed Description

[TableEntry](#).

### 10.303.2 Constructor & Destructor Documentation

#### 10.303.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (
    const char * attribute = 0,
    Type const & type = Type(),
    const char * des = 0 ) [inline]
```

#### 10.303.2.2 ~TableEntry()

```
gdcm::TableEntry::~~TableEntry ( ) [inline]
```

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

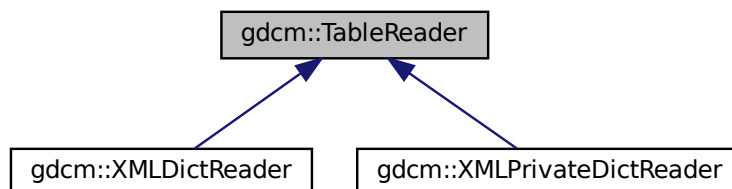
- [gdcmTableEntry.h](#)

## 10.304 gdcm::TableReader Class Reference

Class for representing a [TableReader](#).

```
#include <gdcmTableReader.h>
```

Inheritance diagram for gdcm::TableReader:



## Public Member Functions

- [TableReader](#) (Defs &defs)
- virtual [~TableReader](#) ()
- 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 [HandleIOEntry](#) (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)

### 10.304.1 Detailed Description

Class for representing a [TableReader](#).

#### Note

This class is an empty shell meant to be derived

### 10.304.2 Constructor & Destructor Documentation

#### 10.304.2.1 TableReader()

```
gdcm::TableReader::TableReader (
    Defs & defs ) [inline]
```

#### 10.304.2.2 ~TableReader()

```
virtual gdcm::TableReader::~~TableReader ( ) [inline], [virtual]
```

### 10.304.3 Member Function Documentation

#### 10.304.3.1 CharacterDataHandler()

```
virtual void gdcm::TableReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

#### 10.304.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

#### 10.304.3.3 GetDefs()

```
const Defs& gdcm::TableReader::GetDefs ( ) const [inline]
```

#### 10.304.3.4 GetFilename()

```
const char* gdcm::TableReader::GetFilename ( ) [inline]
```

#### 10.304.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (
    const char ** atts )
```

#### 10.304.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (
    const char ** atts )
```

#### 10.304.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (
    const char ** atts )
```

#### 10.304.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (
    const char ** atts )
```

#### 10.304.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (
    const char ** atts )
```

#### 10.304.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
    const char ** atts )
```

#### 10.304.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
    const char ** atts )
```

**10.304.3.12 HandleModuleEntryDescription()**

```
void gdcm::TableReader::HandleModuleEntryDescription (
    const char ** atts )
```

**10.304.3.13 HandleModuleInclude()**

```
void gdcm::TableReader::HandleModuleInclude (
    const char ** atts )
```

**10.304.3.14 Read()**

```
int gdcm::TableReader::Read ( )
```

**10.304.3.15 SetFilename()**

```
void gdcm::TableReader::SetFilename (
    const char * filename ) [inline]
```

**10.304.3.16 StartElement()**

```
virtual void gdcm::TableReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

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

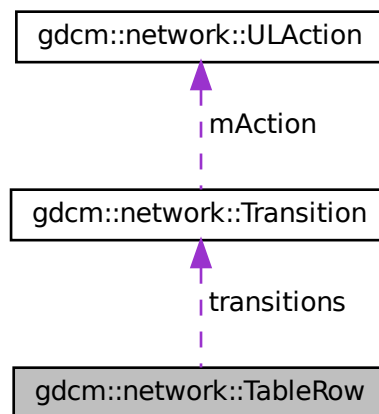
- [gdcmTableReader.h](#)



## 10.305 gdcmm::network::TableRow Class Reference

```
#include <gdcmmULTransitionTable.h>
```

Collaboration diagram for gdcmm::network::TableRow:



### Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

### Public Attributes

- [Transition](#) \* [transitions](#) [[cMaxStateID](#)]

### 10.305.1 Constructor & Destructor Documentation

#### 10.305.1.1 TableRow()

```
gdcmm::network::TableRow::TableRow ( ) [inline]
```

References `gdcmm::network::cMaxStateID`.

### 10.305.1.2 ~TableRow()

```
gdcmm::network::TableRow::~TableRow ( ) [inline]
```

References `gdcmm::network::cMaxStateID`.

## 10.305.2 Member Data Documentation

### 10.305.2.1 transitions

```
Transition* gdcmm::network::TableRow::transitions[cMaxStateID]
```

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

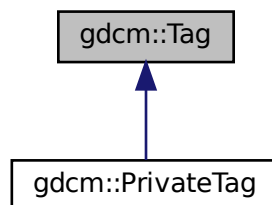
- [gdcmmULTransitionTable.h](#)

## 10.306 gdcmm::Tag Class Reference

Class to represent a DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#)).

```
#include <gdcmmTag.h>
```

Inheritance diagram for `gdcmm::Tag`:



## Public Member Functions

- [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.*
- [Tag](#) (const [Tag](#) &\_val)
- 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
- const uint16\_t & [operator\[\]](#) (const unsigned int &\_id) const  
*Returns the Group or Element of the given Tag, depending on id (0/1)*
- uint16\_t & [operator\[\]](#) (const unsigned int &\_id)  
*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)

### 10.306.1 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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DumpToSQLITE3.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAI BugJPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSEExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndDumpDICOMDIR2.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [rle2img.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), and [VolumeSorter.cxx](#).

### 10.306.2 Constructor & Destructor Documentation

### 10.306.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
    uint16_t group,
    uint16_t element ) [inline]
```

Constructor with 2\*uint16\_t.

### 10.306.2.2 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 `gdcm::operator<<()`, and `gdcm::operator>>()`.

### 10.306.2.3 Tag() [3/3]

```
gdcm::Tag::Tag (
    const Tag & _val ) [inline]
```

References `tag`.

## 10.306.3 Member Function Documentation

### 10.306.3.1 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::DataSet::ComputeGroupLength()`, `IsGroupXX()`, `gdcm::PrivateDict::PrintXML()`, `gdcm::PrivateDict::PrivateTag()`, `gdcm::SequenceOfFragments::ReadValue()`, and `SetPrivateCreator()`.

### 10.306.3.2 GetElementTag()

```
uint32_t gdcm::Tag::GetElementTag ( ) const [inline]
```

Returns the full tag value of the given [Tag](#).

### 10.306.3.3 GetGroup()

```
uint16_t gdcm::Tag::GetGroup ( ) const [inline]
```

Returns the 'Group number' of the given [Tag](#).

#### Examples:

[DuplicatePCDE.cxx](#), and [GenAllVR.cxx](#).

Referenced by `gdcm::DataSet::ComputeGroupLength()`, `gdcm::CommandDataSet::Insert()`, `gdcm::FileMetaInformation::Insert()`, `gdcm::DataSet::Insert()`, `IsGroupXX()`, `gdcm::PrivateDict::PrintXML()`, `gdcm::SequenceOfFragments::ReadValue()`, `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 `SetPrivateCreator()`.

### 10.306.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength ( ) const [inline]
```

return the length of tag (read: size on disk)

### 10.306.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]
```

Return the Private Creator Data [Element](#) tag of a private data element.

References `SetElement()`.

### 10.306.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength ( ) const [inline]
```

return whether the tag correspond to a group length tag:

### 10.306.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
    const Tag & t ) const [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

### 10.306.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal ( ) const [inline]
```

return if the tag is considered to be an illegal tag

### 10.306.3.9 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](#).

Referenced by IsGroupXX(), and SetPrivateCreator().

### 10.306.3.10 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](#).

### 10.306.3.11 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...

### 10.306.3.12 operator!=(())

```
bool gdcm::Tag::operator!= (
    const Tag & _val ) const [inline]
```

References tag.

### 10.306.3.13 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, and tags.

### 10.306.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
    const Tag & t2 ) const [inline]
```



### 10.306.3.15 operator=()

```
Tag& gdcm::Tag::operator= (
    const Tag & _val ) [inline]
```

References tag.

### 10.306.3.16 operator==()

```
bool gdcm::Tag::operator== (
    const Tag & _val ) const [inline]
```

References tag.

### 10.306.3.17 operator[]() [1/2]

```
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)

### 10.306.3.18 operator[]() [2/2]

```
uint16_t& gdcm::Tag::operator[] (
    const unsigned int & _id ) [inline]
```

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

### 10.306.3.19 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)

### 10.306.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcM::Tag::PrintAsContinuousUpperCaseString ( ) const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

### 10.306.3.21 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](#)

### 10.306.3.22 Read()

```
template<typename TSwap >  
std::istream& gdcM::Tag::Read (   
    std::istream & is ) [inline]
```

Read a tag from binary representation.

### 10.306.3.23 ReadFromCommaSeparatedString()

```
bool gdcM::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

### 10.306.3.24 ReadFromContinuousString()

```
bool gdcM::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)

### 10.306.3.25 ReadFromPipeSeparatedString()

```
bool gdcm::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](#)

### 10.306.3.26 SetElement()

```
void gdcm::Tag::SetElement (
    uint16_t element ) [inline]
```

Sets the '[Element](#) number' of the given [Tag](#).

Examples:

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by [GetPrivateCreator\(\)](#), and [gdcm::operator>>\(\)](#).

### 10.306.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (
    uint16_t group,
    uint16_t element ) [inline]
```

Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).

### 10.306.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (
    uint32_t tag ) [inline]
```

Sets the full tag value of the given [Tag](#).

### 10.306.3.29 SetGroup()

```
void gdcM::Tag::SetGroup (
    uint16_t group ) [inline]
```

Sets the 'Group number' of the given [Tag](#).

Referenced by `gdcM::operator>>()`.

### 10.306.3.30 SetPrivateCreator()

```
void gdcM::Tag::SetPrivateCreator (
    Tag const & t ) [inline]
```

Set private creator:

Examples:

[DuplicatePCDE.cxx](#).

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

### 10.306.3.31 Write()

```
template<typename TSwap >
const std::ostream& gdcM::Tag::Write (
    std::ostream & os ) const [inline]
```

Write a tag in binary rep.

Referenced by `gdcM::SequenceOfItems::Write()`, and `gdcM::Item::Write()`.

## 10.306.4 Friends And Related Function Documentation

### 10.306.4.1 operator<<

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

#### 10.306.4.2 operator>>

```
std::istream& operator>> (
    std::istream & _is,
    Tag & _val ) [friend]
```

### 10.306.5 Member Data Documentation

#### 10.306.5.1 bytes

```
char gdcm::Tag::bytes[4]
```

#### 10.306.5.2 tag

```
uint32_t gdcm::Tag::tag
```

Referenced by operator!(), operator<(), operator=(), operator==(), and Tag().

#### 10.306.5.3 tags

```
uint16_t gdcm::Tag::tags[2]
```

Referenced by operator<().

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

- [gdcmTag.h](#)

## 10.307 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.*

### 10.307.1 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](ftp://medical.nema.org/medical/dicom/supps/sup118←_pc.pdf)

### 10.307.2 Constructor & Destructor Documentation

#### 10.307.2.1 TagPath()

```
gdcm::TagPath::TagPath ( )
```

#### 10.307.2.2 ~TagPath()

```
gdcm::TagPath::~~TagPath ( )
```

### 10.307.3 Member Function Documentation

### 10.307.3.1 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

### 10.307.3.2 ConstructFromTagList()

```
bool gdcm::TagPath::ConstructFromTagList (
    Tag const * l,
    unsigned int n )
```

Construct from a list of tags.

### 10.307.3.3 IsValid()

```
static bool gdcm::TagPath::IsValid (
    const char * path ) [static]
```

Return if path is valid or not.

### 10.307.3.4 Print()

```
void gdcm::TagPath::Print (
    std::ostream & ) const
```

### 10.307.3.5 Push() [1/2]

```
bool gdcm::TagPath::Push (
    Tag const & t )
```

### 10.307.3.6 Push() [2/2]

```
bool gdcM::TagPath::Push (
    unsigned int itemnum )
```

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

- [gdcMTagPath.h](#)

## 10.308 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](#) ()
- [~Testing](#) ()
- 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=0)
- static const wchar\_t \* [GetTempDirectoryW](#) (const wchar\_t \*subdir=0)  
*NOT THREAD SAFE.*
- static const char \* [GetTempFilename](#) (const char \*filename, const char \*subdir=0)  
*NOT THREAD SAFE.*
- static const wchar\_t \* [GetTempFilenameW](#) (const wchar\_t \*filename, const wchar\_t \*subdir=0)  
*NOT THREAD SAFE.*

### 10.308.1 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

### 10.308.2 Member Typedef Documentation

#### 10.308.2.1 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

#### 10.308.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType) [2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

### 10.308.3 Constructor & Destructor Documentation

#### 10.308.3.1 Testing()

```
gdcM::Testing::Testing ( ) [inline]
```

#### 10.308.3.2 ~Testing()

```
gdcM::Testing::~~Testing ( ) [inline]
```

### 10.308.4 Member Function Documentation

#### 10.308.4.1 ComputeFileMD5()

```
static bool gdcM::Testing::ComputeFileMD5 (
    const char * filename,
    char digest_str[33] ) [static]
```

#### Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.2 ComputeMD5()

```
static 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.

#### 10.308.4.3 GetDataExtraRoot()

```
static const char* gdcm::Testing::GetDataExtraRoot ( ) [static]
```

Return the GDCM DATA EXTRA ROOT.

Examples:

[DiscriminateVolume.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

#### 10.308.4.4 GetDataRoot()

```
static const char* gdcm::Testing::GetDataRoot ( ) [static]
```

Return the GDCM DATA ROOT.

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [MagnifyFile.cxx](#).

#### 10.308.4.5 GetFileName()

```
static const char* gdcm::Testing::GetFileName (
    unsigned int file ) [static]
```

Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.6 GetFileNames()

```
static const char* const* gdcm::Testing::GetFileNames ( ) [static]
```

return the table of fullpath to gdcmData DICOM files:

Examples:

[TestReader.cxx](#).

#### 10.308.4.7 GetLossyFlagFromFile()

```
static int gdcM::Testing::GetLossyFlagFromFile (
    const char * filepath ) [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

#### 10.308.4.8 GetMD5DataImage()

```
static const char* const* gdcM::Testing::GetMD5DataImage (
    unsigned int file ) [static]
```

#### 10.308.4.9 GetMD5DataImages()

```
static MD5DataImagesType gdcM::Testing::GetMD5DataImages ( ) [static]
```

#### 10.308.4.10 GetMD5FromBrokenFile()

```
static const char* gdcM::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.

#### 10.308.4.11 GetMD5FromFile()

```
static const char* gdcM::Testing::GetMD5FromFile (
    const char * filepath ) [static]
```

#### 10.308.4.12 GetMediaStorageDataFile()

```
static const char* const* gdcM::Testing::GetMediaStorageDataFile (
    unsigned int file ) [static]
```

#### 10.308.4.13 GetMediaStorageDataFiles()

```
static MediaStorageDataFileType gdcm::Testing::GetMediaStorageDataFiles ( ) [static]
```

#### 10.308.4.14 GetMediaStorageFromFile()

```
static const char* gdcm::Testing::GetMediaStorageFromFile (
    const char * filepath ) [static]
```

##### Examples:

[MetaImageMD5Activiz.cs](#), and [TestReader.cxx](#).

#### 10.308.4.15 GetNumberOfFileNames()

```
static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]
```

##### Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.16 GetNumberOfMD5DataImages()

```
static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]
```

#### 10.308.4.17 GetNumberOfMediaStorageDataFiles()

```
static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]
```

#### 10.308.4.18 GetPixelSpacingDataRoot()

```
static const char* gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

**10.308.4.19 GetSelectedPrivateGroupOffsetFromFile()**

```
static 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

**10.308.4.20 GetSelectedTagsOffsetFromFile()**

```
static 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

**10.308.4.21 GetSourceDirectory()**

```
static const char* gdcm::Testing::GetSourceDirectory ( ) [static]
```

Examples:

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

**10.308.4.22 GetStreamOffsetFromFile()**

```
static 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

**10.308.4.23 GetTempDirectory()**

```
static const char* gdcm::Testing::GetTempDirectory (
    const char * subdir = 0 ) [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.24 GetTempDirectoryW()

```
static const wchar_t* gdcm::Testing::GetTempDirectoryW (  
    const wchar_t * subdir = 0 ) [static]
```

NOT THREAD SAFE.

#### 10.308.4.25 GetTempFilename()

```
static const char* gdcm::Testing::GetTempFilename (  
    const char * filename,  
    const char * subdir = 0 ) [static]
```

NOT THREAD SAFE.

Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.308.4.26 GetTempFilenameW()

```
static const wchar_t* gdcm::Testing::GetTempFilenameW (  
    const wchar_t * filename,  
    const wchar_t * subdir = 0 ) [static]
```

NOT THREAD SAFE.

#### 10.308.4.27 Print()

```
void gdcm::Testing::Print (  
    std::ostream & os = std::cout )
```

Print.

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

- [gdcmTesting.h](#)

## 10.309 gdcm::Trace Class Reference

[Trace](#).

```
#include <gdcmTrace.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](#) ()



### 10.309.1 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.

### 10.309.2 Constructor & Destructor Documentation

#### 10.309.2.1 `Trace()`

```
gdcm::Trace::Trace ( )
```

#### 10.309.2.2 `~Trace()`

```
gdcm::Trace::~~Trace ( )
```

### 10.309.3 Member Function Documentation

#### 10.309.3.1 `DebugOff()`

```
static void gdcm::Trace::DebugOff ( ) [static]
```

#### Examples:

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

### 10.309.3.2 DebugOn()

```
static void gdcm::Trace::DebugOn ( ) [static]
```

#### Examples:

[CreateFakePET.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

### 10.309.3.3 ErrorOff()

```
static void gdcm::Trace::ErrorOff ( ) [static]
```

#### Examples:

[MetaImageMD5Activiz.cs](#).

### 10.309.3.4 ErrorOn()

```
static void gdcm::Trace::ErrorOn ( ) [static]
```

### 10.309.3.5 GetDebugFlag()

```
static bool gdcm::Trace::GetDebugFlag ( ) [static]
```

### 10.309.3.6 GetDebugStream()

```
static std::ostream& gdcm::Trace::GetDebugStream ( ) [static]
```

### 10.309.3.7 GetErrorFlag()

```
static bool gdcm::Trace::GetErrorFlag ( ) [static]
```

### 10.309.3.8 GetErrorStream()

```
static std::ostream& gdcm::Trace::GetErrorStream ( ) [static]
```

### 10.309.3.9 GetStream()

```
static std::ostream& gdcm::Trace::GetStream ( ) [static]
```

### 10.309.3.10 GetWarningFlag()

```
static bool gdcm::Trace::GetWarningFlag ( ) [static]
```

### 10.309.3.11 GetWarningStream()

```
static std::ostream& gdcm::Trace::GetWarningStream ( ) [static]
```

### 10.309.3.12 SetDebug()

```
static void gdcm::Trace::SetDebug (
    bool debug ) [static]
```

Turn debug messages on (default: false)

Examples:

[DumpToSQLITE3.cxx](#).

### 10.309.3.13 SetDebugStream()

```
static void gdcm::Trace::SetDebugStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Debug messages:

#### 10.309.3.14 SetError()

```
static void gdcM::Trace::SetError (
    bool debug ) [static]
```

Turn error messages on (default: true)

#### 10.309.3.15 SetErrorStream()

```
static void gdcM::Trace::SetErrorStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Error messages:

Examples:

[CStoreQtProgress.cxx](#).

#### 10.309.3.16 SetStream()

```
static 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:

#### 10.309.3.17 SetStreamToFile()

```
static 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)

#### 10.309.3.18 SetWarning()

```
static void gdcM::Trace::SetWarning (
    bool debug ) [static]
```

Turn warning messages on (default: true)

Examples:

[DumpToSQLITE3.cxx](#).

### 10.309.3.19 SetWarningStream()

```
static void gdcm::Trace::SetWarningStream (
    std::ostream & os ) [static]
```

Explicitly set the stream which receive Warning messages:

### 10.309.3.20 WarningOff()

```
static void gdcm::Trace::WarningOff ( ) [static]
```

Examples:

[MetalImageMD5Activiz.cs](#), and [TestReader.cxx](#).

### 10.309.3.21 WarningOn()

```
static 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](#)

## 10.310 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](#),  
    [TS\\_END](#) }

## Public Member Functions

- [TransferSyntax](#) ([TSType](#) type=[ImplicitVRLittleEndian](#))
- [CanStoreLossy](#) () const
- [NegociatedType](#) [GetNegociatedType](#) () const
- const char \* [GetString](#) () const
- [SwapCode](#) [GetSwapCode](#) () const
- [IsEncapsulated](#) () const
- [IsEncoded](#) () const
- [IsExplicit](#) () const
- [IsImplicit](#) () const
- [IsLossless](#) () const
- [IsLossy](#) () const
- [IsValid](#) () const
- [operator TSType](#) () const

## Static Public Member Functions

- static const char \* [GetTSSString](#) (TSType ts)
- static TSType [GetTSType](#) (const char \*str)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [TransferSyntax](#) &ts)

### 10.310.1 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 [gdcmm::UIDs](#) for a replacement We need: IsSupported 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:

[GetJPEGSamplePrecision.cxx](#), [LargeVRDSExplicit.cxx](#), and [MakeTemplate.cxx](#).

### 10.310.2 Member Enumeration Documentation

#### 10.310.2.1 NegotiatedType

```
enum gdcmm::TransferSyntax::NegociatedType
```

Enumerator

Unknown	
Explicit	
Implicit	

## 10.310.2.2 TSType

```
enum gdcmm::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	
JPIPRreferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	
MPEG4AVCH264BDcompatibleHighProfileLevel4↵ _1	
TS_END	

## 10.310.3 Constructor &amp; Destructor Documentation



### 10.310.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
    TSType type = ImplicitVRLittleEndian ) [inline]
```

## 10.310.4 Member Function Documentation

### 10.310.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy ( ) const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

### 10.310.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const
```

### 10.310.4.3 GetString()

```
const char* gdcm::TransferSyntax::GetString ( ) const [inline]
```

References GetTSString(), and gdcm::operator<<().

### 10.310.4.4 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.

#### 10.310.4.5 GetTSString()

```
static const char* gdcm::TransferSyntax::GetTSString (
    TSType ts ) [static]
```

##### Examples:

[LargeVRDSExplicit.cxx](#).

Referenced by GetString(), and gdcm::operator<<().

#### 10.310.4.6 GetTSType()

```
static TSType gdcm::TransferSyntax::GetTSType (
    const char * str ) [static]
```

#### 10.310.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated ( ) const
```

##### Examples:

[ExtractIconFromFile.cxx](#).

#### 10.310.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded ( ) const
```

#### 10.310.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit ( ) const
```

#### 10.310.4.10 IsImplicit()

```
bool gdcm::TransferSyntax::IsImplicit ( ) const
```

#### 10.310.4.11 IsLossless()

```
bool gdcm::TransferSyntax::IsLossless ( ) const
```

Return true if the transfer syntax algorithm is a lossless algorithm

#### 10.310.4.12 IsLossy()

```
bool gdcm::TransferSyntax::IsLossy ( ) const
```

Return true if the transfer syntax algorithm is a lossy algorithm

#### 10.310.4.13 IsValid()

```
bool gdcm::TransferSyntax::IsValid ( ) const [inline]
```

#### 10.310.4.14 operator TType()

```
gdcm::TransferSyntax::operator TType ( ) const [inline]
```

### 10.310.5 Friends And Related Function Documentation

#### 10.310.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const TransferSyntax & ts ) [friend]
```

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

- [gdcmTransferSyntax.h](#)

## 10.311 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

### 10.311.1 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

### 10.311.2 Constructor & Destructor Documentation

#### 10.311.2.1 TransferSyntaxSub()

```
gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )
```

### 10.311.3 Member Function Documentation

### 10.311.3.1 GetName()

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

References Print(), Read(), SetNameFromUID(), Size(), and Write().

### 10.311.3.2 operator==( )

```
bool gdcm::network::TransferSyntaxSub::operator== (
    const TransferSyntaxSub & ts ) const [inline]
```

### 10.311.3.3 Print()

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

Referenced by GetName().

### 10.311.3.4 Read()

```
std::istream& gdcm::network::TransferSyntaxSub::Read (
    std::istream & is )
```

Referenced by GetName().

### 10.311.3.5 SetName()

```
void gdcm::network::TransferSyntaxSub::SetName (
    const char * name )
```

### 10.311.3.6 SetNameFromUID()

```
void gdcm::network::TransferSyntaxSub::SetNameFromUID (
    UIDs::TSName tsname )
```

Referenced by GetName().

### 10.311.3.7 Size()

```
size_t gdcn::network::TransferSyntaxSub::Size ( ) const
```

Referenced by GetName().

### 10.311.3.8 Write()

```
const std::ostream& gdcn::network::TransferSyntaxSub::Write (
    std::ostream & os ) const
```

Referenced by GetName().

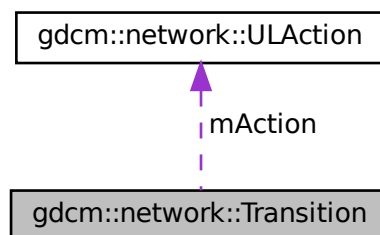
The documentation for this class was generated from the following file:

- [gdcnTransferSyntaxSub.h](#)

## 10.312 gdcn::network::Transition Struct Reference

```
#include <gdcnULTransitionTable.h>
```

Collaboration diagram for gdcn::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](#)

## 10.312.1 Constructor & Destructor Documentation

### 10.312.1.1 [Transition\(\)](#) [1/2]

```
gdcmm::network::Transition::Transition ( ) [inline]
```

References [gdcmm::network::eStaDoesNotExist](#).

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

### 10.312.1.2 [~Transition\(\)](#)

```
gdcmm::network::Transition::~~Transition ( ) [inline]
```

References [mAction](#).

### 10.312.1.3 [Transition\(\)](#) [2/2]

```
gdcmm::network::Transition::Transition (
    int inEndState,
    ULAction * inAction ) [inline]
```

## 10.312.2 Member Function Documentation

### 10.312.2.1 MakeNew()

```
static Transition* gdcM::network::Transition::MakeNew (
    int inEndState,
    ULAction * inAction ) [inline], [static]
```

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

## 10.312.3 Member Data Documentation

### 10.312.3.1 mAction

```
ULAction* gdcM::network::Transition::mAction
```

Referenced by [~Transition\(\)](#).

### 10.312.3.2 mEnd

```
int gdcM::network::Transition::mEnd
```

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

- [gdcMULTransitionTable.h](#)

## 10.313 gdcM::Type Class Reference

[Type](#).

```
#include <gdcMType.h>
```

### Public Types

- enum [Type](#)[Type](#) {  
    [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)

### 10.313.1 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](#).

### 10.313.2 Member Enumeration Documentation

#### 10.313.2.1 TypeType

```
enum gdcmm::Type::TypeType
```

#### Enumerator

T1	
T1C	
T2	
T2C	
T3	
UNKNOWN	

### 10.313.3 Constructor & Destructor Documentation

#### 10.313.3.1 Type()

```
gdcM::Type::Type (
    TypeType type = UNKNOWN ) [inline]
```

### 10.313.4 Member Function Documentation

#### 10.313.4.1 GetTypeString()

```
static const char* gdcM::Type::GetTypeString (
    TypeType type ) [static]
```

Referenced by gdcM::operator<<().

#### 10.313.4.2 GetTypeType()

```
static TypeType gdcM::Type::GetTypeType (
    const char * type ) [static]
```

Referenced by gdcM::ModuleEntry::ModuleEntry().

#### 10.313.4.3 operator TypeType()

```
gdcM::Type::operator TypeType ( ) const [inline]
```

References gdcM::operator<<().

### 10.313.5 Friends And Related Function Documentation

#### 10.313.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Type & vr ) [friend]
```

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

- [gdcmType.h](#)

## 10.314 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

### Public Attributes

- char [Internal](#) [64+1]

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [UI](#) &\_val)

#### 10.314.1 Friends And Related Function Documentation

##### 10.314.1.1 operator<<

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

##### 10.314.2 Member Data Documentation

### 10.314.2.1 Internal

```
char gdcM::UI::Internal[64+1]
```

Referenced by `gdcM::operator<<()`.

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

- [gdcMVR.h](#)

## 10.315 gdcM::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcMUIDGenerator.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)

### 10.315.1 Detailed Description

Class for generating unique UID.

#### Note

bla [Usage](#): 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\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

## 10.315.2 Constructor & Destructor Documentation

### 10.315.2.1 UIDGenerator()

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

By default the root of a UID is a GDCM Root...

## 10.315.3 Member Function Documentation

### 10.315.3.1 Generate()

```
const char* gdcm::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\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [StreamImageReaderTest.cxx](#), [TemplateEmptyImage.cxx](#), and [uid\\_unique.cxx](#).

### 10.315.3.2 GenerateUUID()

```
static bool gdcm::UIDGenerator::GenerateUUID (
    unsigned char * uuid_data ) [static], [protected]
```

### 10.315.3.3 GetGDCMUID()

```
static const char* gdcm::UIDGenerator::GetGDCMUID ( ) [static]
```

Return the default (GDCM) root UID:

#### 10.315.3.4 GetRoot()

```
static const char* gdcm::UIDGenerator::GetRoot ( ) [static]
```

##### Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

#### 10.315.3.5 IsValid()

```
static 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)

#### 10.315.3.6 SetRoot()

```
static 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 responsibility for guaranteeing uniqueness of [UIDs](#) when the root is longer than 26 bytes.

##### Examples:

[ClinicalTrialIdentificationWorkflow.cs](#), [ReformatFile.cs](#), [StandardizeFiles.cs](#), and [uid\\_unique.cxx](#).

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

- [gdcmUIDGenerator.h](#)

## 10.316 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,
  - JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9,
  - JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10,
  - JPEGFullProgressionNonHierarchicalProcess1012Retired = 11,
  - JPEGFullProgressionNonHierarchicalProcess1113Retired = 12,
  - JPEGLosslessNonHierarchicalProcess14 = 13,
  - JPEGLosslessNonHierarchicalProcess15Retired = 14,
  - JPEGExtendedHierarchicalProcess1618Retired = 15,
  - JPEGExtendedHierarchicalProcess1719Retired = 16,
  - JPEGSpectralSelectionHierarchicalProcess2022Retired = 17,
  - JPEGSpectralSelectionHierarchicalProcess2123Retired = 18,
  - JPEGFullProgressionHierarchicalProcess2426Retired = 19,
  - JPEGFullProgressionHierarchicalProcess2527Retired = 20,
  - JPEGLosslessHierarchicalProcess28Retired = 21,
  - JPEGLosslessHierarchicalProcess29Retired = 22,
  - JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↵
  - JPEGImageCompression = 23,
  - JPEGLSLosslessImageCompression = 24,
  - JPEGLSLossyNearLosslessImageCompression = 25,
  - JPEG2000ImageCompressionLosslessOnly = 26,
  - JPEG2000ImageCompression = 27,
  - JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28,
  - JPEG2000Part2MulticomponentImageCompression = 29,
  - JPIPRReferenced = 30,
  - JPIPRReferencedDeflate = 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,  
[SPM2AVG152PDFrameofReference](#) = 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,  
[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,
EnhancedUSVolumeStorage,
SurfaceSegmentationStorage,
BreastTomosynthesisImageStorage }

```

- 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,
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_1_2_4_101,
uid_1_2_840_10008_1_2_4_102,
uid_1_2_840_10008_1_2_4_103 }
```

## Public Member Functions

- 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)

### 10.316.1 Detailed Description

all known uids

Examples:

[GenerateStandardSOPClasses.cxx](#).

### 10.316.2 Member Typedef Documentation

#### 10.316.2.1 TransferSyntaxStringsType

```
typedef const char* const (* gdcm::UIDs::TransferSyntaxStringsType) [2]
```

### 10.316.3 Member Enumeration Documentation

#### 10.316.3.1 TSName

```
enum gdcm::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	
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxfor↔ LosslessJPEGImageCompression	
JPEGLSLosslessImageCompression	
JPEGLSLossyNearLosslessImageCompression	
JPEG2000ImageCompressionLosslessOnly	
JPEG2000ImageCompression	
JPEG2000Part2MulticomponentImageCompressionLosslessOnly	
JPEG2000Part2MulticomponentImageCompression	
JPIPReferenced	
JPIPReferencedDeflate	
MPEG2MainProfileMainLevel	
RLELossless	
RFC2557MIMEencapsulation	
XMLEncoding	
MediaStorageDirectoryStorage	
TalairachBrainAtlasFrameofReference	
SPM2T1FrameofReference	
SPM2T2FrameofReference	
SPM2PDFrameofReference	
SPM2EPIFrameofReference	
SPM2FILT1FrameofReference	
SPM2PETFrameofReference	
SPM2TRANSMFrameofReference	

## Enumerator

SPM2SPECTFrameofReference	
SPM2GRAYFrameofReference	
SPM2WHITEFrameofReference	
SPM2CSFFFrameofReference	
SPM2BRAINMASKFrameofReference	
SPM2AVG305T1FrameofReference	
SPM2AVG152T1FrameofReference	
SPM2AVG152T2FrameofReference	
SPM2AVG152PDFrameofReference	
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	

## Enumerator

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	

## Enumerator

StandaloneCurveStorageRetired	
WaveformStorageTrialRetired	
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	

## Enumerator

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	

## Enumerator

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	

## Enumerator

dicomDevice	
dicomNetworkAE	
dicomNetworkConnection	
dicomUniqueAETitle	
dicomTransferCapability	
VLWholeSlideMicroscopyImageStorage	
EnhancedUSVolumeStorage	
SurfaceSegmentationStorage	
BreastTomosynthesisImageStorage	

## 10.316.3.2 TSType

```
enum gdcm::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	



## Enumerator

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	

## Enumerator

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	

## Enumerator

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	

## Enumerator

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	

## Enumerator

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	

## Enumerator

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_1_2_4_101	
uid_1_2_840_10008_1_2_4_102	
uid_1_2_840_10008_1_2_4_103	

## 10.316.4 Member Function Documentation

## 10.316.4.1 GetName()

```
const char* gdcm::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 `gdcm::operator<<()`.

#### 10.316.4.2 GetNumberOfTransferSyntaxStrings()

```
static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]
```

#### 10.316.4.3 GetString()

```
const char* gdcm::UIDs::GetString ( ) const
```

When object is Initialize function return the uid return NULL when not initialized

Examples:

[GenerateStandardSOPClasses.cxx](#).

Referenced by `gdcm::operator<<()`.

#### 10.316.4.4 GetTransferSyntaxString()

```
static const char* const* gdcm::UIDs::GetTransferSyntaxString (
    unsigned int ts ) [static]
```

#### 10.316.4.5 GetTransferSyntaxStrings()

```
static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]
```

#### 10.316.4.6 GetUIDName()

```
static const char* gdcm::UIDs::GetUIDName (
    unsigned int ts ) [static]
```

#### 10.316.4.7 GetUIDString()

```
static const char* gdcm::UIDs::GetUIDString (
    unsigned int ts ) [static]
```

#### 10.316.4.8 operator TType()

```
gdcm::UIDs::operator TType ( ) const [inline]
```

#### 10.316.4.9 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](#)

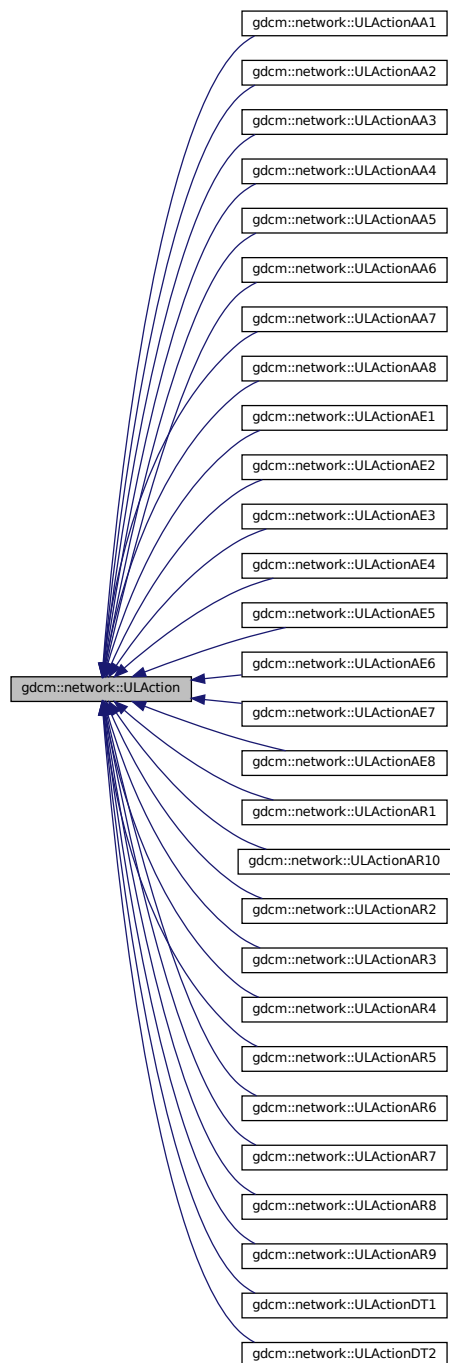
## 10.317 gdcm::network::ULAction Class Reference

[ULAction](#).

```
#include <gdcmULAction.h>
```



Inheritance diagram for gdcmm::network::ULAction:



## Public Member Functions

- [ULAction](#) ()

- virtual `~ULAction ()`
- virtual `EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaiting, ForEvent, EEventID &outRaisedEvent)=0`

### 10.317.1 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 `ULEvent` (ie, what happened as a result of the action).

This `ULEvent` 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 rest of the state transitions can happen.

### 10.317.2 Constructor & Destructor Documentation

#### 10.317.2.1 ULAction()

```
gdcmm::network::ULAction::ULAction ( ) [inline]
```

## 10.317.2.2 ~ULAction()

```
virtual gdcm::network::ULAction::~~ULAction ( ) [inline], [virtual]
```

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

## 10.317.3 Member Function Documentation

## 10.317.3.1 PerformAction()

```
virtual EStateID gdcm::network::ULAction::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [pure virtual]
```

Implemented in [gdcm::network::ULActionAR10](#), [gdcm::network::ULActionAR9](#), [gdcm::network::ULActionAE8](#), [gdcm::network::ULActionAA8](#), [gdcm::network::ULActionAR8](#), [gdcm::network::ULActionAE7](#), [gdcm::network::ULActionAA7](#), [gdcm::network::ULActionAR7](#), [gdcm::network::ULActionAE6](#), [gdcm::network::ULActionAA6](#), [gdcm::network::ULActionAR6](#), [gdcm::network::ULActionAA5](#), [gdcm::network::ULActionAE5](#), [gdcm::network::ULActionAR5](#), [gdcm::network::ULActionAA4](#), [gdcm::network::ULActionAE4](#), [gdcm::network::ULActionAR4](#), [gdcm::network::ULActionAA3](#), [gdcm::network::ULActionAE3](#), [gdcm::network::ULActionAR3](#), [gdcm::network::ULActionAA2](#), [gdcm::network::ULActionAE2](#), [gdcm::network::ULActionAR2](#), [gdcm::network::ULActionDT2](#), [gdcm::network::ULActionAA1](#), [gdcm::network::ULActionAE1](#), [gdcm::network::ULActionAR1](#), and [gdcm::network::ULActionDT1](#).

Referenced by [~ULAction\(\)](#).

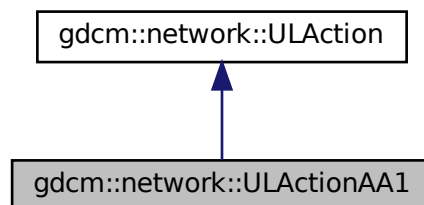
The documentation for this class was generated from the following file:

- [gdcmULAction.h](#)

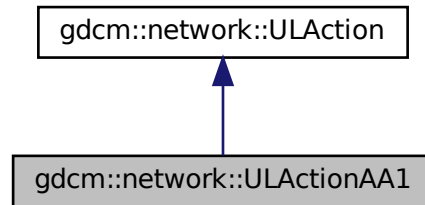
## 10.318 gdcm::network::ULActionAA1 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.318.1 Member Function Documentation

### 10.318.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

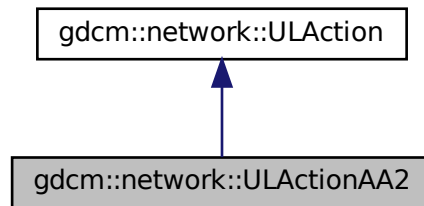
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

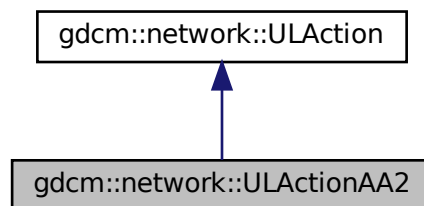
## 10.319 gdcmm::network::ULActionAA2 Class Reference

```
#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 &outWaitingFor↔  
Event, [EEventID](#) &outRaisedEvent)

### 10.319.1 Member Function Documentation

### 10.319.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

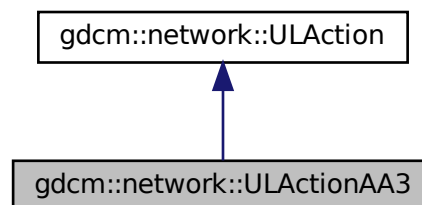
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

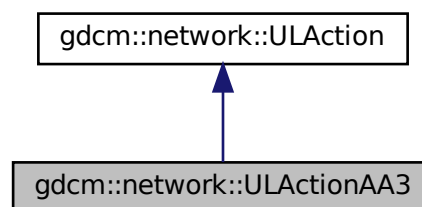
## 10.320 gdcmm::network::ULActionAA3 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA3:



Collaboration diagram for gdcmm::network::ULActionAA3:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.320.1 Member Function Documentation

#### 10.320.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

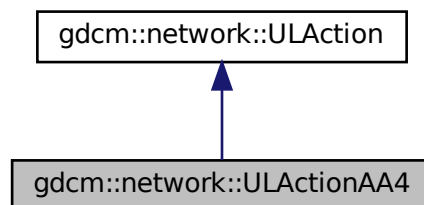
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

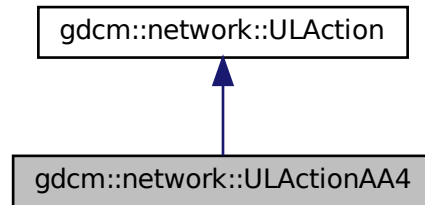
## 10.321 gdcm::network::ULActionAA4 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for `gdcm::network::ULActionAA4`:



Collaboration diagram for `gdcm::network::ULActionAA4`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.321.1 Member Function Documentation

#### 10.321.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

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

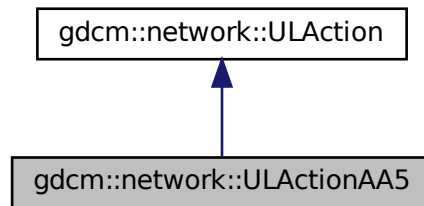
- [gdcmULActionAA.h](#)



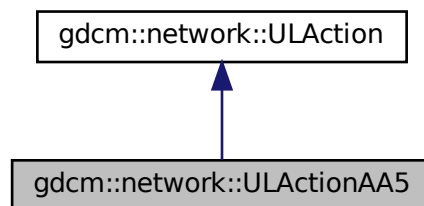
## 10.322 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for gdcm::network::ULActionAA5:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↔  
Event, [EEventID](#) &outRaisedEvent)

### 10.322.1 Member Function Documentation

### 10.322.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

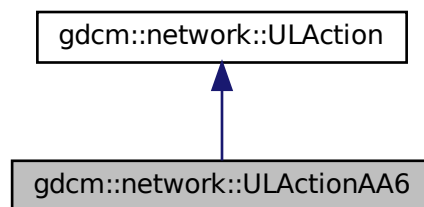
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

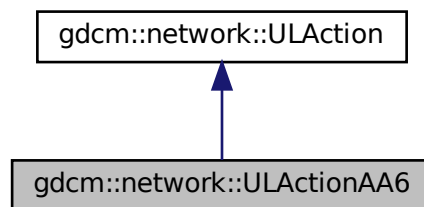
## 10.323 gdcmm::network::ULActionAA6 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA6:



Collaboration diagram for gdcmm::network::ULActionAA6:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.323.1 Member Function Documentation

#### 10.323.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

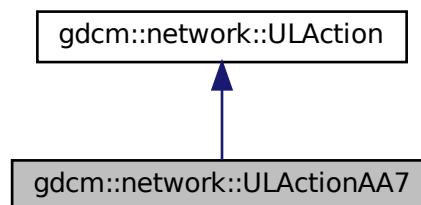
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

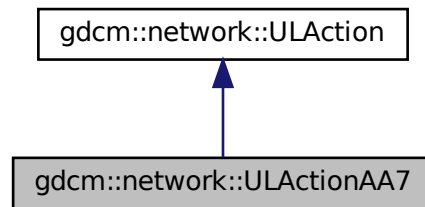
## 10.324 gdcm::network::ULActionAA7 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.324.1 Member Function Documentation

### 10.324.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAA7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

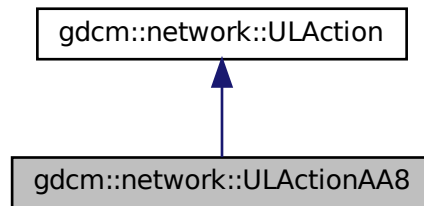
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

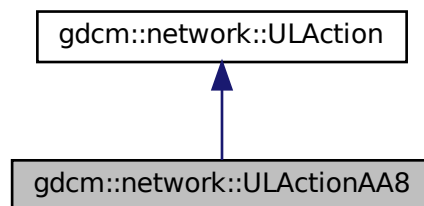
## 10.325 gdcmm::network::ULActionAA8 Class Reference

```
#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 &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 10.325.1 Member Function Documentation

### 10.325.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAA8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

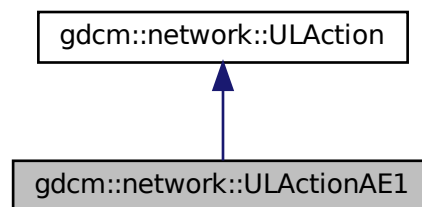
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

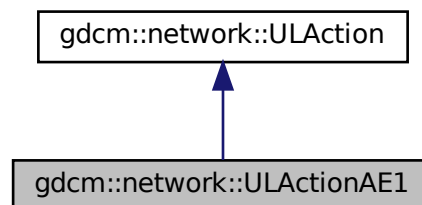
## 10.326 gdcmm::network::ULActionAE1 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE1:



Collaboration diagram for gdcmm::network::ULActionAE1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.326.1 Member Function Documentation

#### 10.326.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

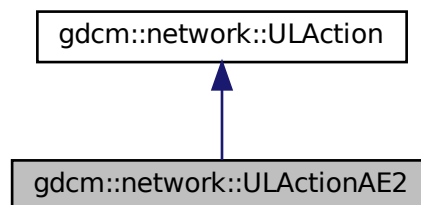
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

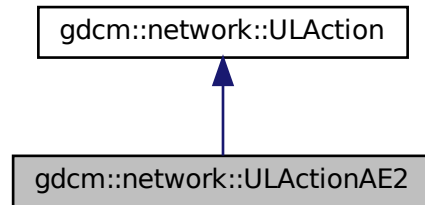
## 10.327 gdcm::network::ULActionAE2 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE2`:



Collaboration diagram for `gdcm::network::ULActionAE2`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.327.1 Member Function Documentation

### 10.327.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

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

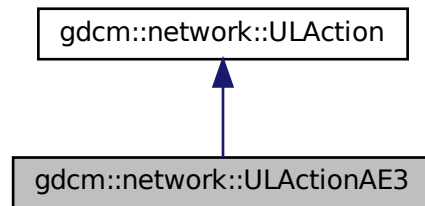
- [gdcmULActionAE.h](#)



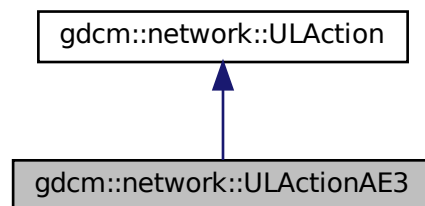
## 10.328 gdcmm::network::ULActionAE3 Class Reference

```
#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 &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 10.328.1 Member Function Documentation

### 10.328.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

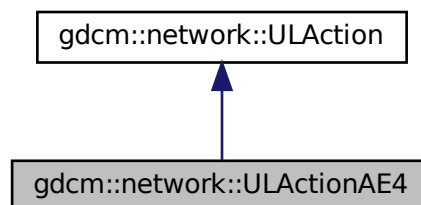
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

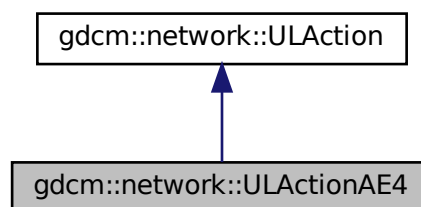
## 10.329 gdcmm::network::ULActionAE4 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE4:



Collaboration diagram for gdcmm::network::ULActionAE4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.329.1 Member Function Documentation

#### 10.329.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

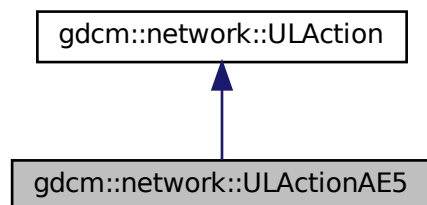
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

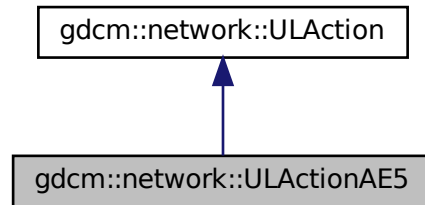
## 10.330 gdcm::network::ULActionAE5 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.330.1 Member Function Documentation

### 10.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

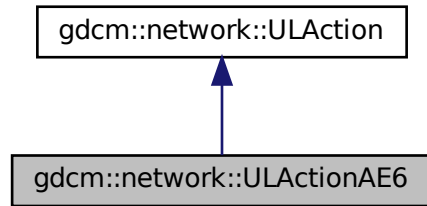
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

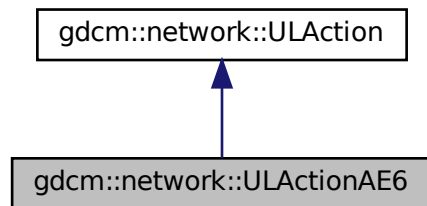
## 10.331 gdcm::network::ULActionAE6 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE6:



Collaboration diagram for gdcm::network::ULActionAE6:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 10.331.1 Member Function Documentation

### 10.331.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAE6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

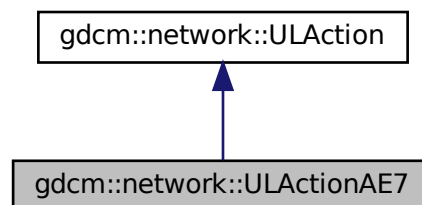
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

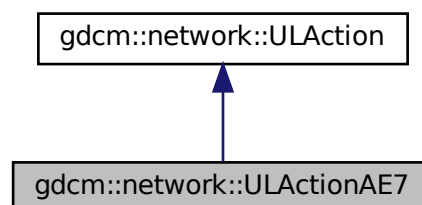
## 10.332 gdcmm::network::ULActionAE7 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE7:



Collaboration diagram for gdcmm::network::ULActionAE7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.332.1 Member Function Documentation

#### 10.332.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

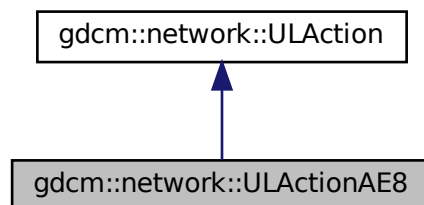
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

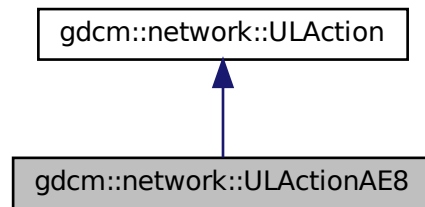
## 10.333 gdcm::network::ULActionAE8 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE8`:



Collaboration diagram for `gdcm::network::ULActionAE8`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.333.1 Member Function Documentation

### 10.333.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAE8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

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

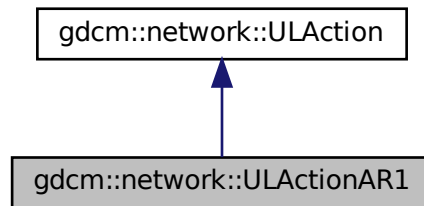
- [gdcmULActionAE.h](#)



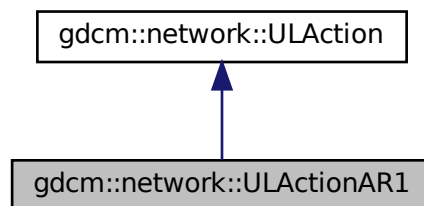
## 10.334 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcm::network::ULActionAR1:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↔  
Event, [EEventID](#) &outRaisedEvent)

#### 10.334.1 Member Function Documentation

### 10.334.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

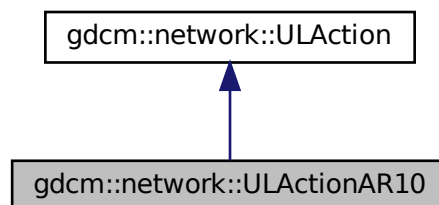
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

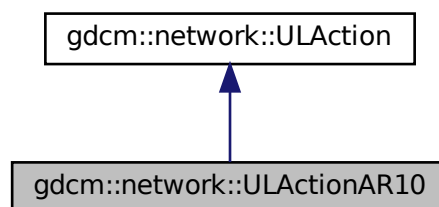
## 10.335 gdcmm::network::ULActionAR10 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR10:



Collaboration diagram for gdcmm::network::ULActionAR10:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.335.1 Member Function Documentation

#### 10.335.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

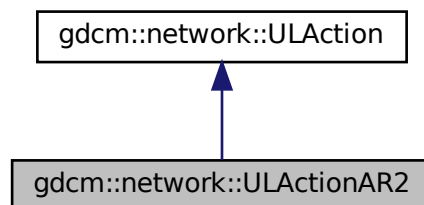
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

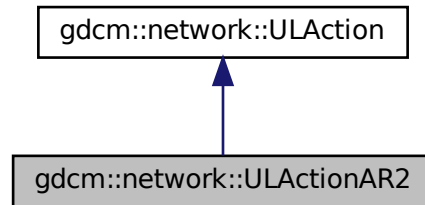
## 10.336 gdcm::network::ULActionAR2 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR2`:



Collaboration diagram for `gdcm::network::ULActionAR2`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.336.1 Member Function Documentation

### 10.336.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

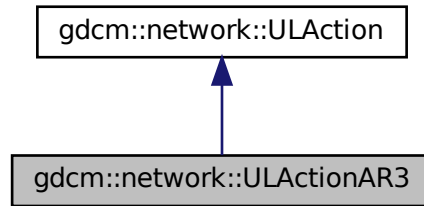
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

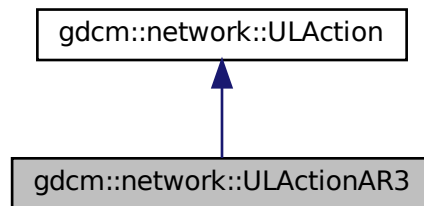
## 10.337 gdcmm::network::ULActionAR3 Class Reference

```
#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 &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 10.337.1 Member Function Documentation

### 10.337.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR3::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

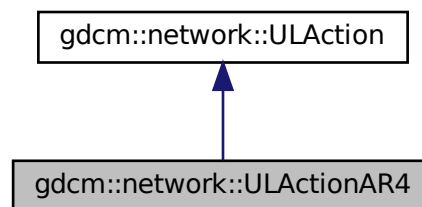
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

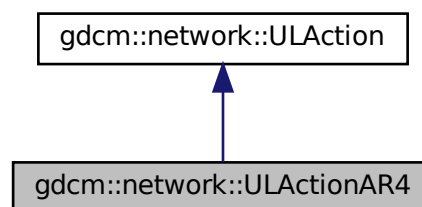
## 10.338 gdcmm::network::ULActionAR4 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR4:



Collaboration diagram for gdcmm::network::ULActionAR4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.338.1 Member Function Documentation

#### 10.338.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

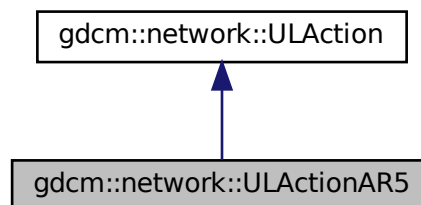
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

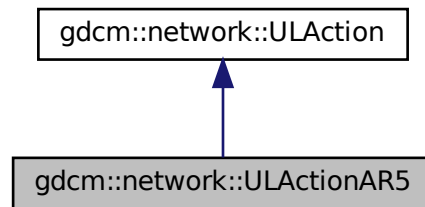
## 10.339 gdcm::network::ULActionAR5 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR5`:



Collaboration diagram for `gdcm::network::ULActionAR5`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 10.339.1 Member Function Documentation

### 10.339.1.1 PerformAction()

```

EStateID gdcm::network::ULActionAR5::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
  
```

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

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

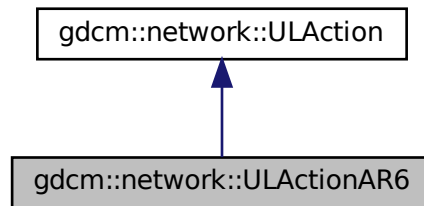
- [gdcmULActionAR.h](#)



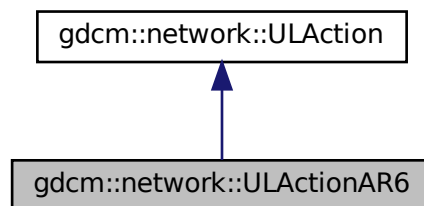
## 10.340 gdcmm::network::ULActionAR6 Class Reference

```
#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 &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 10.340.1 Member Function Documentation

### 10.340.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR6::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

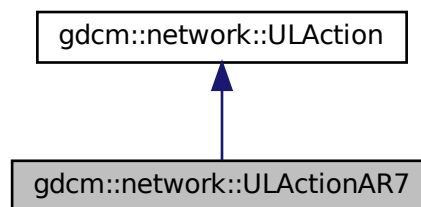
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

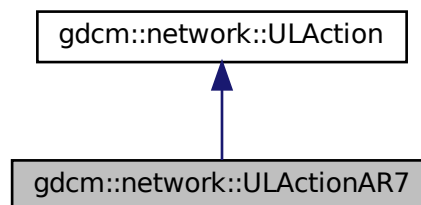
## 10.341 gdcmm::network::ULActionAR7 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR7:



Collaboration diagram for gdcmm::network::ULActionAR7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.341.1 Member Function Documentation

#### 10.341.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

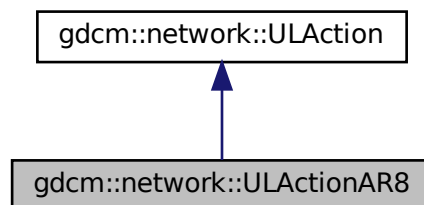
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

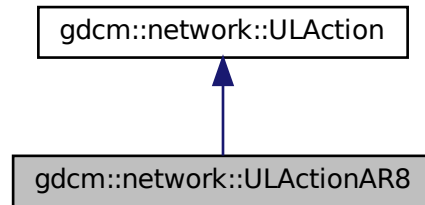
## 10.342 gdcm::network::ULActionAR8 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR8:



Collaboration diagram for `gdcm::network::ULActionAR8`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.342.1 Member Function Documentation

#### 10.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR8::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

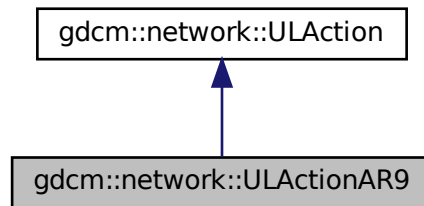
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

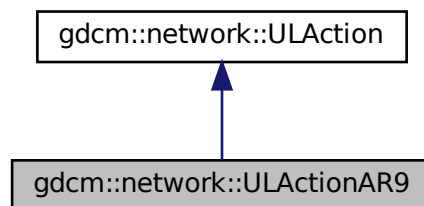
## 10.343 gdcmm::network::ULActionAR9 Class Reference

```
#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 &outWaitingFor↔  
Event, [EEventID](#) &outRaisedEvent)

### 10.343.1 Member Function Documentation

### 10.343.1.1 PerformAction()

```
EStateID gdcmm::network::ULActionAR9::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

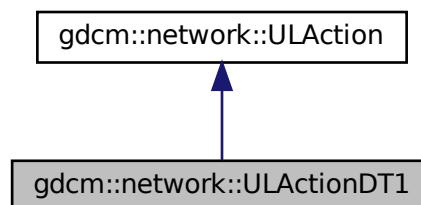
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

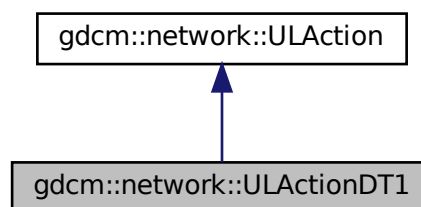
## 10.344 gdcmm::network::ULActionDT1 Class Reference

```
#include <gdcmmULActionDT.h>
```

Inheritance diagram for gdcmm::network::ULActionDT1:



Collaboration diagram for gdcmm::network::ULActionDT1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.344.1 Member Function Documentation

#### 10.344.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

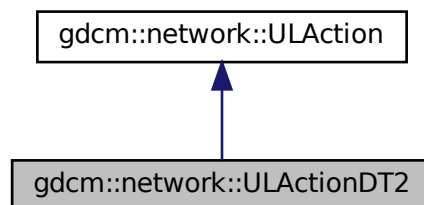
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

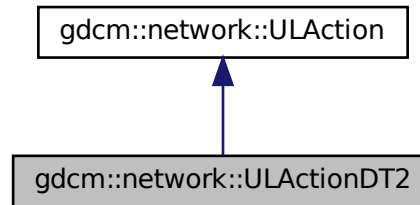
## 10.345 gdcm::network::ULActionDT2 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for `gdcm::network::ULActionDT2`:



Collaboration diagram for `gdcm::network::ULActionDT2`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 10.345.1 Member Function Documentation

#### 10.345.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT2::PerformAction (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) [virtual]
```

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

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

- [gdcmULActionDT.h](#)

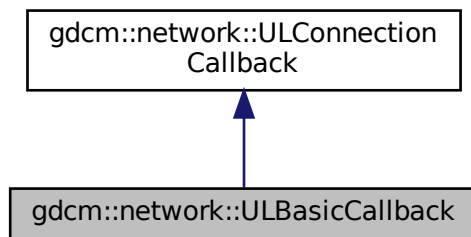


## 10.346 gdcm::network::ULBasicCallback Class Reference

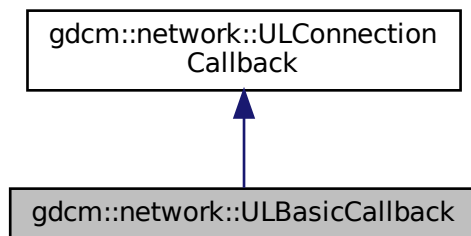
[ULBasicCallback](#).

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



### Public Member Functions

- [ULBasicCallback](#) ()
- virtual [~ULBasicCallback](#) ()
- std::vector< [DataSet](#) > const & [GetDataSets](#) () const
- std::vector< [DataSet](#) > const & [GetResponses](#) () const
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)

## Additional Inherited Members

### 10.346.1 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](#).

### 10.346.2 Constructor & Destructor Documentation

#### 10.346.2.1 ULBasicCallback()

```
gdcmm::network::ULBasicCallback::ULBasicCallback ( ) [inline]
```

#### 10.346.2.2 ~ULBasicCallback()

```
virtual gdcmm::network::ULBasicCallback::~~ULBasicCallback ( ) [inline], [virtual]
```

### 10.346.3 Member Function Documentation

#### 10.346.3.1 GetDataSets()

```
std::vector<DataSet> const& gdcmm::network::ULBasicCallback::GetDataSets ( ) const
```

#### 10.346.3.2 GetResponses()

```
std::vector<DataSet> const& gdcmm::network::ULBasicCallback::GetResponses ( ) const
```

## 10.346.3.3 HandleDataSet()

```
virtual void gdcm::network::ULBasicCallback::HandleDataSet (
    const DataSet & inDataSet ) [virtual]
```

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

## 10.346.3.4 HandleResponse()

```
virtual void gdcm::network::ULBasicCallback::HandleResponse (
    const DataSet & inDataSet ) [virtual]
```

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

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

- [gdcmULBasicCallback.h](#)

## 10.347 gdcm::network::ULConnection Class Reference

[ULConnection](#).

```
#include <gdcmULConnection.h>
```

## Public Member Functions

- [ULConnection](#) (const [ULConnectionInfo](#) &inUserInformation)
- virtual [~ULConnection](#) ()
- void [AddAcceptedPresentationContext](#) (const [PresentationContextAC](#) &inPC)
- [PresentationContextRQ FindContext](#) (const [DataElement](#) &de) const
- std::vector< [PresentationContextAC](#) > const & [GetAcceptedPresentationContexts](#) () const
- std::vector< [PresentationContextAC](#) > & [GetAcceptedPresentationContexts](#) ()
- 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 [SetMaxPDUSize](#) (uint32\_t inSize)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContextRQ](#) > &inContexts)
- void [SetPresentationContexts](#) (const std::vector< [PresentationContext](#) > &inContexts)
- void [SetState](#) (const [EStateID](#) &inState)
- void [StopProtocol](#) ()

## Friends

- class [ULActionAE6](#)
- class [ULConnectionManager](#)

### 10.347.1 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.

### 10.347.2 Constructor & Destructor Documentation

#### 10.347.2.1 [ULConnection\(\)](#)

```
gdcm::network::ULConnection::ULConnection (
    const ULConnectionInfo & inUserInformation )
```

#### 10.347.2.2 [~ULConnection\(\)](#)

```
virtual gdcm::network::ULConnection::~~ULConnection ( ) [virtual]
```

### 10.347.3 Member Function Documentation

### 10.347.3.1 AddAcceptedPresentationContext()

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
    const PresentationContextAC & inPC )
```

### 10.347.3.2 FindContext()

```
PresentationContextRQ gdcm::network::ULConnection::FindContext (
    const DataElement & de ) const
```

### 10.347.3.3 GetAcceptedPresentationContexts() [1/2]

```
std::vector<PresentationContextAC> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( ) const
```

### 10.347.3.4 GetAcceptedPresentationContexts() [2/2]

```
std::vector<PresentationContextAC>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( )
```

### 10.347.3.5 GetConnectionInfo()

```
const ULConnectionInfo& gdcm::network::ULConnection::GetConnectionInfo ( ) const
```

### 10.347.3.6 GetMaxPDUSize()

```
uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const
```

### 10.347.3.7 GetPresentationContextACByID()

```
const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID (
    uint8_t id ) const
```

**10.347.3.8 GetPresentationContextIDFromPresentationContext()**

```
uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
    PresentationContextRQ const & pc ) const
```

return 0 upon error

**10.347.3.9 GetPresentationContextRQByID()**

```
const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (
    uint8_t id ) const
```

**10.347.3.10 GetPresentationContexts()**

```
std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts (
) const
```

**10.347.3.11 GetProtocol()**

```
std::iostream* gdcm::network::ULConnection::GetProtocol ( )
```

**10.347.3.12 GetState()**

```
EStateID gdcm::network::ULConnection::GetState ( ) const
```

**10.347.3.13 GetTimer()**

```
ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )
```

#### 10.347.3.14 InitializeConnection()

```
bool gdcmm::network::ULConnection::InitializeConnection ( )
```

used to establish scu connections

#### 10.347.3.15 InitializeIncomingConnection()

```
bool gdcmm::network::ULConnection::InitializeIncomingConnection ( )
```

used to establish scp connections

#### 10.347.3.16 SetMaxPDUSize()

```
void gdcmm::network::ULConnection::SetMaxPDUSize (
    uint32_t inSize )
```

#### 10.347.3.17 SetPresentationContexts() [1/2]

```
void gdcmm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContextRQ > & inContexts )
```

#### 10.347.3.18 SetPresentationContexts() [2/2]

```
void gdcmm::network::ULConnection::SetPresentationContexts (
    const std::vector< PresentationContext > & inContexts )
```

#### 10.347.3.19 SetState()

```
void gdcmm::network::ULConnection::SetState (
    const EStateID & inState )
```

### 10.347.3.20 StopProtocol()

```
void gdcm::network::ULConnection::StopProtocol ( )
```

## 10.347.4 Friends And Related Function Documentation

### 10.347.4.1 ULActionAE6

```
friend class ULActionAE6 [friend]
```

### 10.347.4.2 ULConnectionManager

```
friend class ULConnectionManager [friend]
```

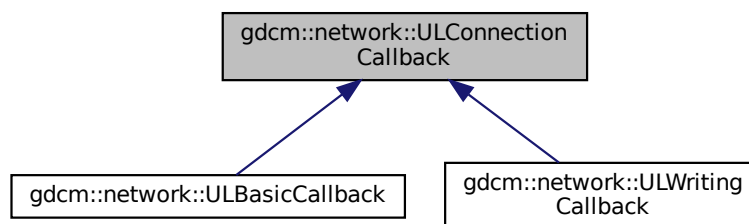
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

## 10.348 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:





## Public Member Functions

- [ULConnectionCallback](#) ()
- virtual [~ULConnectionCallback](#) ()
- 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](#)

### 10.348.1 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.

### 10.348.2 Constructor & Destructor Documentation

#### 10.348.2.1 [ULConnectionCallback](#)()

```
gdcm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]
```

#### 10.348.2.2 [~ULConnectionCallback](#)()

```
virtual gdcm::network::ULConnectionCallback::~~ULConnectionCallback ( ) [inline], [virtual]
```

### 10.348.3 Member Function Documentation

#### 10.348.3.1 DataSetHandled()

```
void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline], [protected]
```

#### 10.348.3.2 DataSetHandles()

```
bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]
```

#### 10.348.3.3 HandleDataSet()

```
virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

#### 10.348.3.4 HandleResponse()

```
virtual void gdcm::network::ULConnectionCallback::HandleResponse (
    const DataSet & inDataSet ) [pure virtual]
```

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

#### 10.348.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]
```

#### 10.348.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
    const bool imp ) [inline]
```

## 10.348.4 Member Data Documentation

### 10.348.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit [protected]
```

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

- [gdcmULConnectionCallback.h](#)

## 10.349 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](#) ([UserInfo](#) const &inUserInfo, const char \*inCalledAETitle, const char \*inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

### 10.349.1 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.

### 10.349.2 Constructor & Destructor Documentation

#### 10.349.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ( )
```

### 10.349.3 Member Function Documentation

#### 10.349.3.1 GetCalledAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const
```

#### 10.349.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const
```

#### 10.349.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const
```

#### 10.349.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const
```

#### 10.349.3.5 GetCallingAETitle()

```
const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const
```

#### 10.349.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const
```

### 10.349.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (
    UserInformation const & inUserInformation,
    const char * inCalledAETitle,
    const char * inCallingAETitle,
    unsigned long inCalledIPAddress,
    int inCalledIPPort,
    std::string inCalledComputerName )
```

### 10.349.3.8 SetMaxPDULength()

```
void gdcm::network::ULConnectionInfo::SetMaxPDULength (
    unsigned long inMaxPDULength )
```

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

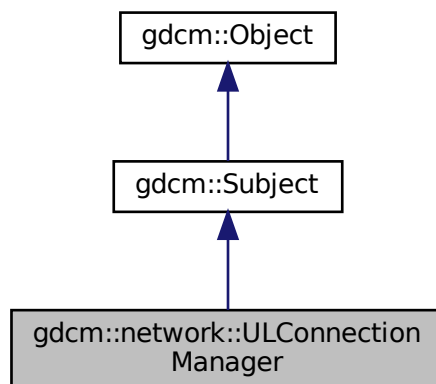
- [gdcmULConnectionInfo.h](#)

## 10.350 gdcm::network::ULConnectionManager Class Reference

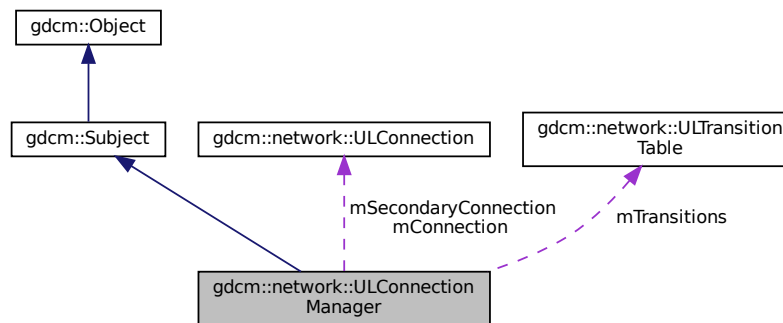
[ULConnectionManager](#).

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for `gdcmm::network::ULConnectionManager`:



## Public Member Functions

- [ULConnectionManager](#) ()
- virtual [~ULConnectionManager](#) ()
- 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=NULL, std::streampos dataSetOffset=0)
- void [SendStore](#) (const [File](#) &file, [ULConnectionCallback](#) \*inCallback, std::istream \*pStream=NULL, std::streampos dataSetOffset=0)  
*callback based API*

## Protected Member Functions

- [ULConnectionManager](#) (const [ULConnectionManager](#) &inCM)
- [EStateID RunEventLoop](#) ([ULError](#) &inEvent, [ULConnection](#) \*inWhichConnection, [ULConnectionCallback](#) \*inCallback, const bool &startWaiting)
- [EStateID RunMoveEventLoop](#) ([ULError](#) &inEvent, [ULConnectionCallback](#) \*inCallback)

## Protected Attributes

- [ULConnection](#) \* mConnection
- [ULConnection](#) \* mSecondaryConnection
- [ULTransitionTable](#) mTransitions

### 10.350.1 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.

### 10.350.2 Constructor & Destructor Documentation

#### 10.350.2.1 [ULConnectionManager](#)() [1/2]

```
gdcmm::network::ULConnectionManager::ULConnectionManager (
    const ULConnectionManager & inCM ) [protected]
```

#### 10.350.2.2 [ULConnectionManager](#)() [2/2]

```
gdcmm::network::ULConnectionManager::ULConnectionManager ( )
```

#### 10.350.2.3 [~ULConnectionManager](#)()

```
virtual gdcmm::network::ULConnectionManager::~~ULConnectionManager ( ) [virtual]
```

### 10.350.3 Member Function Documentation

#### 10.350.3.1 BreakConnection()

```
bool gdcm::network::ULConnectionManager::BreakConnection (
    const double & inTimeout )
```

#### 10.350.3.2 BreakConnectionNow()

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ( )
```

#### 10.350.3.3 EstablishConnection()

```
bool gdcm::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.

#### 10.350.3.4 EstablishConnectionMove()

```
bool gdcm::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



### 10.350.3.5 RunEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunEventLoop (
    ULEvent & inEvent,
    ULConnection * inWhichConnection,
    ULConnectionCallback * inCallback,
    const bool & startWaiting ) [protected]
```

### 10.350.3.6 RunMoveEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (
    ULEvent & inEvent,
    ULConnectionCallback * inCallback ) [protected]
```

### 10.350.3.7 SendEcho()

```
std::vector<PresentationDataValue> gdcm::network::ULConnectionManager::SendEcho ( )
```

### 10.350.3.8 SendFind() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery )
```

### 10.350.3.9 SendFind() [2/2]

```
void gdcm::network::ULConnectionManager::SendFind (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

### 10.350.3.10 SendMove() [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery )
```

**10.350.3.11 SendMove()** [2/2]

```
bool gdcm::network::ULConnectionManager::SendMove (
    const BaseRootQuery * inRootQuery,
    ULConnectionCallback * inCallback )
```

return false upon error

**10.350.3.12 SendNAction()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery )
```

**10.350.3.13 SendNAction()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNAction (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.14 SendNCreate()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery )
```

**10.350.3.15 SendNCreate()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.16 SendNDelete()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery )
```

**10.350.3.17 SendNDelete()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNDelete (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.18 SendNEventReport()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery )
```

**10.350.3.19 SendNEventReport()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNEventReport (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.20 SendNGet()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery )
```

**10.350.3.21 SendNGet()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.22 SendNSet()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery )
```

**10.350.3.23 SendNSet()** [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
    const BaseQuery * inQuery,
    ULConnectionCallback * inCallback )
```

**10.350.3.24 SendStore()** [1/2]

```
std::vector<DataSet> gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    std::istream * pStream = NULL,
    std::streampos dataSetOffset = 0 )
```

**10.350.3.25 SendStore()** [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
    const File & file,
    ULConnectionCallback * inCallback,
    std::istream * pStream = NULL,
    std::streampos dataSetOffset = 0 )
```

callback based API

**10.350.4 Member Data Documentation****10.350.4.1 mConnection**

`ULConnection*` gdcm::network::ULConnectionManager::mConnection [protected]

**10.350.4.2 mSecondaryConnection**

`ULConnection*` gdcm::network::ULConnectionManager::mSecondaryConnection [protected]

## 10.350.4.3 mTransitions

```
ULTransitionTable gdcmm::network::ULConnectionManager::mTransitions [protected]
```

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

- [gdcmmULConnectionManager.h](#)

## 10.351 gdcmm::network::ULEvent Class Reference

[ULEvent](#).

```
#include <gdcmmULEvent.h>
```

## Public Member Functions

- [ULEvent](#) (const [EEventID](#) &inEventID, std::vector< [BasePDU](#) \*> inBasePDU, std::istream \*iStream=NULL, std::streampos posDataSet=0)
- [ULEvent](#) (const [EEventID](#) &inEventID, [BasePDU](#) \*inBasePDU, std::istream \*iStream=NULL, 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)

## 10.351.1 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!

## 10.351.2 Constructor &amp; Destructor Documentation

**10.351.2.1 ULEvent()** [1/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    std::vector< BasePDU *> inBasePDU,
    std::istream * iStream = NULL,
    std::streampos posDataSet = 0 ) [inline]
```

**10.351.2.2 ULEvent()** [2/2]

```
gdcm::network::ULEvent::ULEvent (
    const EEventID & inEventID,
    BasePDU * inBasePDU,
    std::istream * iStream = NULL,
    std::streampos posDataSet = 0 ) [inline]
```

**10.351.2.3 ~ULEvent()**

```
gdcm::network::ULEvent::~~ULEvent ( ) [inline]
```

**10.351.3 Member Function Documentation****10.351.3.1 GetDataSetPos()**

```
std::streampos gdcm::network::ULEvent::GetDataSetPos ( ) const [inline]
```

**10.351.3.2 GetEvent()**

```
EEventID gdcm::network::ULEvent::GetEvent ( ) const [inline]
```

**10.351.3.3 GetIStream()**

```
std::istream* gdcm::network::ULEvent::GetIStream ( ) const [inline]
```

## 10.351.3.4 GetPDUs()

```
std::vector<BasePDU*> const& gdcmm::network::ULEvent::GetPDUs ( ) const [inline]
```

## 10.351.3.5 SetEvent()

```
void gdcmm::network::ULEvent::SetEvent (
    const EEventID & inEvent ) [inline]
```

## 10.351.3.6 SetPDU()

```
void gdcmm::network::ULEvent::SetPDU (
    std::vector< BasePDU *> const & inPDU ) [inline]
```

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

- [gdcmmULEvent.h](#)

## 10.352 gdcmm::network::ULTransitionTable Class Reference

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmmULTransitionTable.h>
```

## Public Member Functions

- [ULTransitionTable](#) ()
- void [HandleEvent](#) (Subject \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

## 10.352.1 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.

## 10.352.2 Constructor & Destructor Documentation

### 10.352.2.1 ULTransitionTable()

```
gdcmm::network::ULTransitionTable::ULTransitionTable ( )
```

## 10.352.3 Member Function Documentation

### 10.352.3.1 HandleEvent()

```
void gdcmm::network::ULTransitionTable::HandleEvent (
    Subject * s,
    ULEvent & inEvent,
    ULConnection & inConnection,
    bool & outWaitingForEvent,
    EEventID & outRaisedEvent ) const
```

### 10.352.3.2 PrintTable()

```
void gdcmm::network::ULTransitionTable::PrintTable ( ) const
```

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

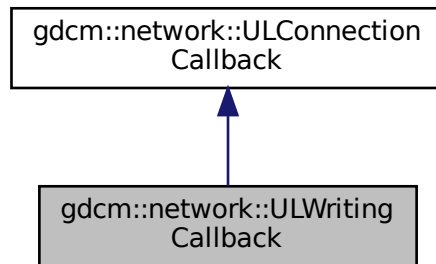
- [gdcmmULTransitionTable.h](#)



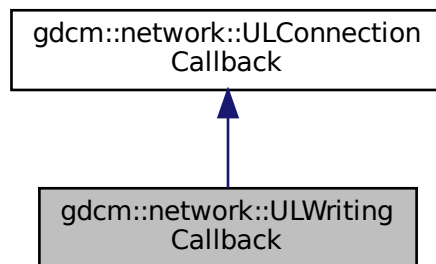
## 10.353 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



### Public Member Functions

- [ULWritingCallback](#) ()
- virtual [~ULWritingCallback](#) ()
- virtual void [HandleDataSet](#) (const [DataSet](#) &inDataSet)
- virtual void [HandleResponse](#) (const [DataSet](#) &inDataSet)
- void [SetDirectory](#) (const std::string &inDirectoryName)

*provide the directory into which all files are written.*

## Additional Inherited Members

### 10.353.1 Constructor & Destructor Documentation

#### 10.353.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback ( ) [inline]
```

#### 10.353.1.2 ~ULWritingCallback()

```
virtual gdcm::network::ULWritingCallback::~~ULWritingCallback ( ) [inline], [virtual]
```

### 10.353.2 Member Function Documentation

#### 10.353.2.1 HandleDataSet()

```
virtual void gdcm::network::ULWritingCallback::HandleDataSet (
    const DataSet & inDataSet ) [virtual]
```

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

#### 10.353.2.2 HandleResponse()

```
virtual void gdcm::network::ULWritingCallback::HandleResponse (
    const DataSet & inDataSet ) [virtual]
```

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

### 10.353.2.3 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:

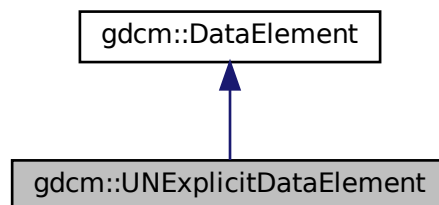
- [gdcmULWritingCallback.h](#)

## 10.354 gdcm::UNExplicitDataElement Class Reference

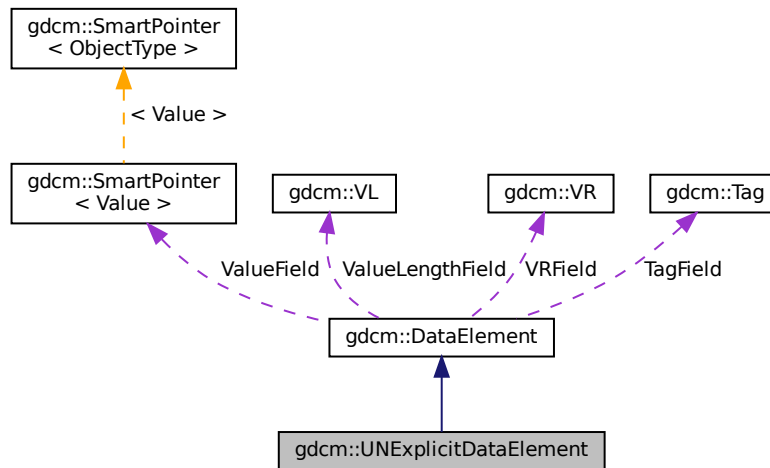
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#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)

## Additional Inherited Members

### 10.354.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

#### Note

bla

### 10.354.2 Member Function Documentation

### 10.354.2.1 GetLength()

```
VL gdcm::UNExplicitDataElement::GetLength ( ) const
```

### 10.354.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::UNExplicitDataElement::Read (   
    std::istream & is )
```

### 10.354.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::UNExplicitDataElement::ReadPreValue (   
    std::istream & is )
```

### 10.354.2.4 ReadValue()

```
template<typename TSwap >  
std::istream& gdcm::UNExplicitDataElement::ReadValue (   
    std::istream & is,   
    bool readvalues = true )
```

### 10.354.2.5 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:

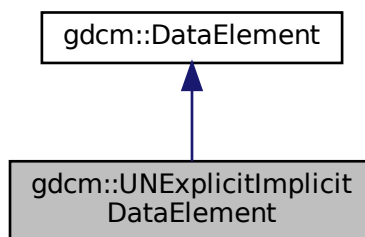
- [gdcmUNExplicitDataElement.h](#)

## 10.355 gdcm::UNExplicitImplicitDataElement Class Reference

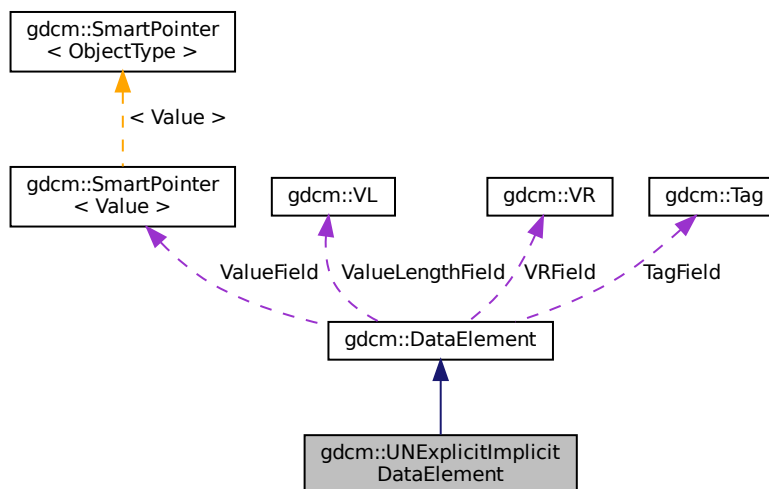
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#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)`

## Additional Inherited Members

### 10.355.1 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](#)

### 10.355.2 Member Function Documentation

#### 10.355.2.1 GetLength()

```
VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const
```

#### 10.355.2.2 Read()

```
template<typename TSwap >  
std::istream& gdcm::UNExplicitImplicitDataElement::Read (  
    std::istream & is )
```

#### 10.355.2.3 ReadPreValue()

```
template<typename TSwap >  
std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (  
    std::istream & is )
```

#### 10.355.2.4 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](#)

## 10.356 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)

#### 10.356.1 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](#)

#### 10.356.2 Member Function Documentation



### 10.356.2.1 Pack()

```
static 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  $(n / 2) * 3$

### 10.356.2.2 Unpack()

```
static bool gdcm::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:

- [gdcmUnpacker12Bits.h](#)

## 10.357 gdcm::Usage Class Reference

[Usage.](#)

```
#include <gdcmUsage.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)

## 10.357.1 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 MAN↔  
DATORY 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.

## 10.357.2 Member Enumeration Documentation

### 10.357.2.1 UsageType

```
enum gdcm::Usage::UsageType
```

#### Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

### 10.357.3 Constructor & Destructor Documentation

#### 10.357.3.1 Usage()

```
gdcm::Usage::Usage (
    UsageType type = Invalid ) [inline]
```

### 10.357.4 Member Function Documentation

#### 10.357.4.1 GetUsageString()

```
static const char* gdcm::Usage::GetUsageString (
    UsageType type ) [static]
```

Referenced by `gdcm::operator<<()`.

#### 10.357.4.2 GetUsageType()

```
static UsageType gdcm::Usage::GetUsageType (
    const char * type ) [static]
```

#### 10.357.4.3 operator UsageType()

```
gdcm::Usage::operator UsageType ( ) const [inline]
```

References `gdcm::operator<<()`.

### 10.357.5 Friends And Related Function Documentation

### 10.357.5.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & os,  
    const Usage & vr ) [friend]
```

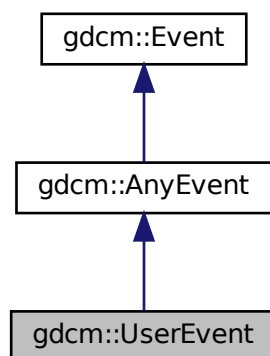
The documentation for this class was generated from the following file:

- [gdcmUsage.h](#)

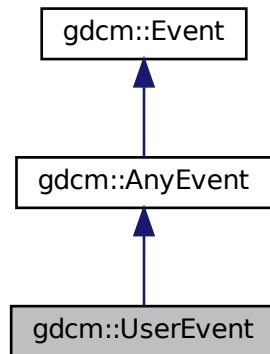
## 10.358 gdcm::UserEvent Class Reference

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

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 10.359 gdcm::network::UserInfo Class Reference

[UserInfo](#).

```
#include <gdcmUserInfo.h>
```

### Public Member Functions

- [UserInfo](#) ()
- [~UserInfo](#) ()
- void [AddRoleSelectionSub](#) ([RoleSelectionSub](#) const &r)
- void [AddSOPClassExtendedNegociationSub](#) ([SOPClassExtendedNegociationSub](#) const &s)
- const [MaximumLengthSub](#) & [GetMaximumLengthSub](#) () const
- [MaximumLengthSub](#) & [GetMaximumLengthSub](#) ()
- [UserInfo](#) & [operator=](#) (const [UserInfo](#) &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 10.359.1 Detailed Description

[UserInformation](#).

[Table 9-16](#) USER INFORMATION ITEM FIELDS

TODO what is the goal of :

[Table 9-20](#) USER INFORMATION ITEM FIELDS

### 10.359.2 Constructor & Destructor Documentation

#### 10.359.2.1 UserInformation()

```
gdcmm::network::UserInformation::UserInformation ( )
```

Referenced by GetMaximumLengthSub().

#### 10.359.2.2 ~UserInformation()

```
gdcmm::network::UserInformation::~~UserInformation ( )
```

### 10.359.3 Member Function Documentation

#### 10.359.3.1 AddRoleSelectionSub()

```
void gdcmm::network::UserInformation::AddRoleSelectionSub (
    RoleSelectionSub const & r )
```

Referenced by GetMaximumLengthSub().

### 10.359.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInfo::AddSOPClassExtendedNegociationSub (
    SOPClassExtendedNegociationSub const & s )
```

Referenced by GetMaximumLengthSub().

### 10.359.3.3 GetMaximumLengthSub() [1/2]

```
const MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( ) const [inline]
```

### 10.359.3.4 GetMaximumLengthSub() [2/2]

```
MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( ) [inline]
```

References AddRoleSelectionSub(), AddSOPClassExtendedNegociationSub(), operator=(), and UserInfo().

### 10.359.3.5 operator=()

```
UserInfo& gdcm::network::UserInfo::operator= (
    const UserInfo & )
```

Referenced by GetMaximumLengthSub().

### 10.359.3.6 Print()

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

### 10.359.3.7 Read()

```
std::istream& gdcm::network::UserInfo::Read (
    std::istream & is )
```

#### 10.359.3.8 Size()

```
size_t gdcn::network::UserInformation::Size ( ) const
```

#### 10.359.3.9 Write()

```
const std::ostream& gdcn::network::UserInformation::Write (
    std::ostream & os ) const
```

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

- [gdcnUserInformation.h](#)

## 10.360 gdcn::UUIDGenerator Class Reference

Class for generating unique UUID.

```
#include <gdcnUUIDGenerator.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.*

#### 10.360.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

#### 10.360.2 Member Function Documentation



### 10.360.2.1 Generate()

```
const char* gdcm::UUIDGenerator::Generate ( )
```

Return the generated uuid NOT THREAD SAFE

### 10.360.2.2 IsValid()

```
static 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:

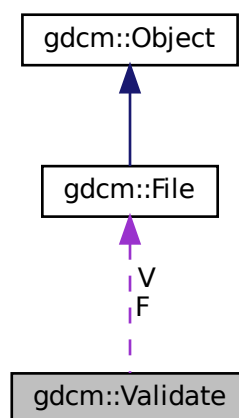
- [gdcmUUIDGenerator.h](#)

## 10.361 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](#)

### 10.361.1 Detailed Description

[Validate](#) class.

### 10.361.2 Constructor & Destructor Documentation

#### 10.361.2.1 [Validate](#)()

```
gdcmm::Validate::Validate ( )
```

#### 10.361.2.2 [~Validate](#)()

```
gdcmm::Validate::~~Validate ( )
```

### 10.361.3 Member Function Documentation

#### 10.361.3.1 [GetValidatedFile](#)()

```
const File& gdcmm::Validate::GetValidatedFile ( ) [inline]
```

### 10.361.3.2 SetFile()

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

### 10.361.3.3 Validation()

```
void gdcm::Validate::Validation ( )
```

## 10.361.4 Member Data Documentation

### 10.361.4.1 F

```
const File* gdcm::Validate::F [protected]
```

### 10.361.4.2 V

```
File gdcm::Validate::V [protected]
```

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

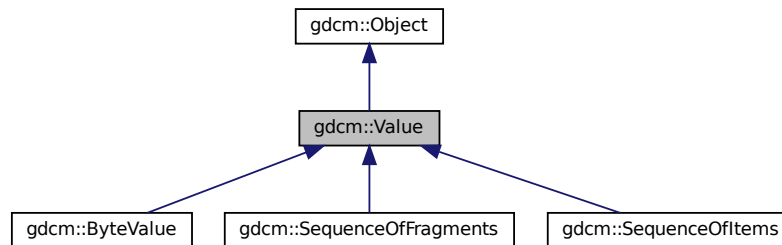
- [gdcmValidate.h](#)

## 10.362 gdcm::Value Class Reference

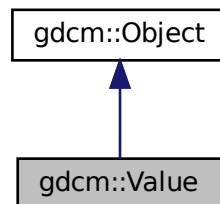
Class to represent the value of a Data [Element](#).

```
#include <gdcmValue.h>
```

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



### Public Member Functions

- [Value](#) ()
- [~Value](#) ()
- virtual void [Clear](#) ()=0
- virtual [VL GetLength](#) () const =0
- virtual bool [operator==](#) (const [Value](#) &val) const =0
- virtual void [SetLength](#) ([VL](#) l)=0

### Protected Member Functions

- virtual void [SetLengthOnly](#) ([VL](#) l)

## Friends

- class [DataElement](#)

### 10.362.1 Detailed Description

Class to represent the value of a Data [Element](#).

#### Note

VALUE: A component of a [Value](#) Field. A [Value](#) Field may consist of one or more of these components.

### 10.362.2 Constructor & Destructor Documentation

#### 10.362.2.1 Value()

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

#### 10.362.2.2 ~Value()

```
gdcm::Value::~~Value ( ) [inline]
```

References [gdcm::operator==\(.\)](#).

### 10.362.3 Member Function Documentation

#### 10.362.3.1 Clear()

```
virtual void gdcm::Value::Clear ( ) [pure virtual]
```

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

### 10.362.3.2 GetLength()

```
virtual VL gdcM::Value::GetLength ( ) const [pure virtual]
```

Implemented in [gdcM::ByteValue](#), [gdcM::SequenceOfItems](#), and [gdcM::SequenceOfFragments](#).

Referenced by [gdcM::DataSet::InsertDataElement\(\)](#), and [gdcM::DataElement::SetValue\(\)](#).

### 10.362.3.3 operator==( )

```
virtual bool gdcM::Value::operator== (
    const Value & val ) const [pure virtual]
```

Implemented in [gdcM::SequenceOfFragments](#), [gdcM::SequenceOfItems](#), and [gdcM::ByteValue](#).

### 10.362.3.4 SetLength()

```
virtual void gdcM::Value::SetLength (
    VL l ) [pure virtual]
```

Implemented in [gdcM::ByteValue](#), [gdcM::SequenceOfItems](#), and [gdcM::SequenceOfFragments](#).

### 10.362.3.5 SetLengthOnly()

```
virtual void gdcM::Value::SetLengthOnly (
    VL l ) [protected], [virtual]
```

Reimplemented in [gdcM::ByteValue](#).

## 10.362.4 Friends And Related Function Documentation

### 10.362.4.1 DataElement

```
friend class DataElement [friend]
```

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

- [gdcMValue.h](#)

## 10.363 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)

### 10.363.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>  
class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

### 10.363.2 Member Function Documentation

#### 10.363.2.1 Read()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>  
static std::istream& gdcm::ValueIO< TDE, TSwap, TType >::Read (  
    std::istream & is,  
    Value & v,  
    bool readvalues ) [static]
```

#### 10.363.2.2 Write()

```
template<typename TDE , typename TSwap , typename TType = uint8_t>  
static const std::ostream& gdcm::ValueIO< TDE, TSwap, TType >::Write (  
    std::ostream & os,  
    const Value & v ) [static]
```

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

- [gdcmValueIO.h](#)

## 10.364 gdcM::MrProtocol::Vector3 Struct Reference

```
#include <gdcMMrProtocol.h>
```

### Public Attributes

- double [dCor](#)
- double [dSag](#)
- double [dTra](#)

### 10.364.1 Member Data Documentation

#### 10.364.1.1 dCor

```
double gdcM::MrProtocol::Vector3::dCor
```

#### 10.364.1.2 dSag

```
double gdcM::MrProtocol::Vector3::dSag
```

#### 10.364.1.3 dTra

```
double gdcM::MrProtocol::Vector3::dTra
```

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

- [gdcMMrProtocol.h](#)

## 10.365 gdcM::Version Class Reference

major/minor and build version

```
#include <gdcMVersion.h>
```



## Public Member Functions

- [Version](#) ()
- [~Version](#) ()
- 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)

### 10.365.1 Detailed Description

major/minor and build version

### 10.365.2 Constructor & Destructor Documentation

#### 10.365.2.1 Version()

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

#### 10.365.2.2 ~Version()

```
gdcm::Version::~~Version ( ) [inline]
```

### 10.365.3 Member Function Documentation

#### 10.365.3.1 GetBuildVersion()

```
static int gdcM::Version::GetBuildVersion ( ) [static]
```

#### 10.365.3.2 GetMajorVersion()

```
static int gdcM::Version::GetMajorVersion ( ) [static]
```

#### 10.365.3.3 GetMinorVersion()

```
static int gdcM::Version::GetMinorVersion ( ) [static]
```

#### 10.365.3.4 GetVersion()

```
static const char* gdcM::Version::GetVersion ( ) [static]
```

#### 10.365.3.5 Print()

```
void gdcM::Version::Print (
    std::ostream & os = std::cout ) const
```

Referenced by `gdcM::operator<<()`.

### 10.365.4 Friends And Related Function Documentation

#### 10.365.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & _os,
    const Version & v ) [friend]
```

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

- [gdcMVersion.h](#)

## 10.366 gdcm::VL Class Reference

Value Length.

```
#include <gdcmVL.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)

### 10.366.1 Detailed Description

[Value](#) Length.

#### Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

#### Examples:

[ReadAndDumpDICOMDIR2.cxx](#), and [rle2img.cxx](#).

### 10.366.2 Member Typedef Documentation

#### 10.366.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```

### 10.366.3 Constructor & Destructor Documentation

#### 10.366.3.1 VL()

```
gdcm::VL::VL (
    uint32_t vl = 0 ) [inline]
```

### 10.366.4 Member Function Documentation

#### 10.366.4.1 GetLength()

```
VL gdcm::VL::GetLength ( ) const [inline]
```

#### Examples:

[ReadAndDumpDICOMDIR2.cxx](#).

References `gdcm::operator<<()`.

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, and `gdcm::Item::Write()`.

#### 10.366.4.2 GetVL16Max()

```
static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]
```

#### 10.366.4.3 GetVL32Max()

```
static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]
```

#### 10.366.4.4 IsOdd()

```
bool gdcm::VL::IsOdd ( ) const [inline]
```

Return whether or not the [VL](#) is odd or not.

#### 10.366.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined ( ) const [inline]
```

#### 10.366.4.6 operator uint32\_t()

```
gdcm::VL::operator uint32_t ( ) const [inline]
```

#### 10.366.4.7 operator++() [1/2]

```
VL& gdcm::VL::operator++ ( ) [inline]
```

#### 10.366.4.8 operator++() [2/2]

```
VL gdcm::VL::operator++ (
    int ) [inline]
```

**10.366.4.9 operator+=()**

```
VL& gdcM::VL::operator+= (
    VL const & vl ) [inline]
```

**+= operator**

**10.366.4.10 Read()**

```
template<typename TSwap >
std::istream& gdcM::VL::Read (
    std::istream & is ) [inline]
```

**10.366.4.11 Read16()**

```
template<typename TSwap >
std::istream& gdcM::VL::Read16 (
    std::istream & is ) [inline]
```

**10.366.4.12 SetToUndefined()**

```
void gdcM::VL::SetToUndefined ( ) [inline]
```

**10.366.4.13 Write()**

```
template<typename TSwap >
const std::ostream& gdcM::VL::Write (
    std::ostream & os ) const [inline]
```

Referenced by `gdcM::Fragment::Write()`, `gdcM::SequenceOfItems::Write()`, `gdcM::Item::Write()`, and `gdcM::SequenceOfFragments::Write()`.

**10.366.4.14 Write16()**

```
template<typename TSwap >
const std::ostream& gdcM::VL::Write16 (
    std::ostream & os ) const [inline]
```

## 10.366.5 Friends And Related Function Documentation

### 10.366.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VL & vl ) [friend]
```

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

- [gdcmVL.h](#)

## 10.367 gdcm::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 <gdcmVM.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,
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](#) (unsigned int 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)

## 10.367.1 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



## 10.367.2 Member Enumeration Documentation

### 10.367.2.1 VMType

enum `gdcmm::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	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

### 10.367.3 Constructor & Destructor Documentation

#### 10.367.3.1 VM()

```
gdcM::VM::VM (
    VMType type = VM0 ) [inline]
```

### 10.367.4 Member Function Documentation

#### 10.367.4.1 Compatible()

```
bool gdcM::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.

#### 10.367.4.2 GetIndex()

```
static unsigned int gdcM::VM::GetIndex (
    VMType vm ) [static], [protected]
```

#### 10.367.4.3 GetLength()

```
unsigned int gdcM::VM::GetLength ( ) const
```

#### 10.367.4.4 GetNumberOfElementsFromArray()

```
static size_t gdcM::VM::GetNumberOfElementsFromArray (
    const char * array,
    size_t length ) [static]
```

#### 10.367.4.5 GetVMString()

```
static 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 `gdcm::operator<<()`.

#### 10.367.4.6 GetVMType()

```
static VMType gdcm::VM::GetVMType (
    const char * vm ) [static]
```

#### 10.367.4.7 GetVMTypeFromLength()

```
static VMType gdcm::VM::GetVMTypeFromLength (
    unsigned int length,
    unsigned int size ) [static]
```

#### 10.367.4.8 IsValid()

```
static 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

#### 10.367.4.9 operator VMType()

```
gdcm::VM::operator VMType ( ) const [inline]
```

References `gdcm::operator<<()`.

### 10.367.5 Friends And Related Function Documentation

### 10.367.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VM & vm ) [friend]
```

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

- [gdcmmVM.h](#)

## 10.368 gdcmm::VMToLength< T > Struct Template Reference

```
#include <gdcmmVM.h>
```

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

- [gdcmmVM.h](#)

## 10.369 gdcmm::VR Class Reference

[VR](#) class.

```
#include <gdcmmVR.h>
```

### Public Types

- enum [VRType](#) {  
    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,  
    OW = 16384,  
    PN = 32768,

```

SH = 65536,
SL = 131072,
SQ = 262144,
SS = 524288,
ST = 1048576,
TM = 2097152,
UI = 4194304,
UL = 8388608,
UN = 16777216,
US = 33554432,
UT = 67108864,
OB_OW = OB | OW,
US_SS = US | SS,
US_SS_OW = US | SS | 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 | SQ | UN | UT,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UI | UT,
VRBINARY = AT | FL | FD | OB | OD | OF | OL | OW | SL | SQ | SS | UL | UN | US,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OD | OW | OB | UN,
VRALL = VRASCII | VRBINARY,
VR_END = OL+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)`

## 10.369.1 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](#), and [GenFakelIdentifyFile.cxx](#).

## 10.369.2 Member Enumeration Documentation

### 10.369.2.1 VRType

```
enum gdcm::VR::VRType
```

#### Enumerator

INVALID	
AE	
AS	
AT	
CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	

## Enumerator

OL	
OW	
PN	
SH	
SL	
SQ	
SS	
ST	
TM	
UI	
UL	
UN	
US	
UT	
OB_OW	
US_SS	
US_SS_OW	
VL16	
VL32	
VRASCII	
VRBINARY	
VR_VM1	
VRALL	
VR_END	

## Examples:

[NewSequence.cs](#).

### 10.369.3 Constructor & Destructor Documentation

#### 10.369.3.1 VR()

```
gdcm::VR::VR (
    VRType vr = INVALID ) [inline]
```

### 10.369.4 Member Function Documentation

#### 10.369.4.1 CanDisplay()

```
static bool gdcm::VR::CanDisplay (
    VRType vr ) [static]
```

#### 10.369.4.2 Compatible()

```
bool gdcm::VR::Compatible (
    VR const & vr ) const
```

#### 10.369.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength ( ) const [inline]
```

#### Examples:

[ReadAndDumpDICOMDIR2.cxx](#).

#### 10.369.4.4 GetLength() [2/2]

```
static uint32_t gdcm::VR::GetLength (
    VRType vr ) [inline], [static]
```

#### 10.369.4.5 GetSize()

```
unsigned int gdcm::VR::GetSize ( ) const [inline]
```

References INVALID, OB\_OW, US\_SS, US\_SS\_OW, VL16, VL32, VR\_END, VR\_VM1, VRALL, VRASCII, VRBINARY, and VRTypeTemplateCase.

#### 10.369.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof ( ) const
```



#### 10.369.4.7 GetVRString()

```
static const char* gdcm::VR::GetVRString (
    VRType vr ) [static]
```

Referenced by gdcm::operator<<().

#### 10.369.4.8 GetVRStringFromFile()

```
static const char* gdcm::VR::GetVRStringFromFile (
    VRType vr ) [static]
```

#### 10.369.4.9 GetVRType()

```
static VRType gdcm::VR::GetVRType (
    const char * vr ) [static]
```

#### 10.369.4.10 GetVRTypeFromFile()

```
static VRType gdcm::VR::GetVRTypeFromFile (
    const char * vr ) [static]
```

#### 10.369.4.11 IsASCII()

```
static bool gdcm::VR::IsASCII (
    VRType vr ) [static]
```

#### 10.369.4.12 IsASCII2()

```
static bool gdcm::VR::IsASCII2 (
    VRType vr ) [static]
```

**10.369.4.13 IsBinary()**

```
static bool gdcm::VR::IsBinary (
    VRType vr ) [static]
```

**10.369.4.14 IsBinary2()**

```
static bool gdcm::VR::IsBinary2 (
    VRType vr ) [static]
```

**10.369.4.15 IsDual()**

```
bool gdcm::VR::IsDual ( ) const
```

**10.369.4.16 IsSwap()**

```
static bool gdcm::VR::IsSwap (
    const char * vr ) [static]
```

**10.369.4.17 IsValid()** [1/2]

```
static bool gdcm::VR::IsValid (
    const char * vr ) [static]
```

**10.369.4.18 IsValid()** [2/2]

```
static bool gdcm::VR::IsValid (
    const char * vr1,
    VRType vr2 ) [static]
```

#### 10.369.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile ( ) const
```

Referenced by `gdcm::DataElement::SetVR()`.

#### 10.369.4.20 operator VRType()

```
gdcm::VR::operator VRType ( ) const [inline]
```

#### 10.369.4.21 Read()

```
std::istream& gdcm::VR::Read (
    std::istream & is ) [inline]
```

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

#### 10.369.4.22 Write()

```
const std::ostream& gdcm::VR::Write (
    std::ostream & os ) const [inline]
```

References `gdcmAssertAlwaysMacro`, `INVALID`, and `gdcm::operator<<()`.

### 10.369.5 Friends And Related Function Documentation

#### 10.369.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const VR & vr ) [friend]
```

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

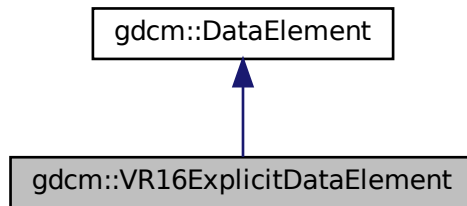
- [gdcmVR.h](#)

## 10.370 gdcm::VR16ExplicitDataElement Class Reference

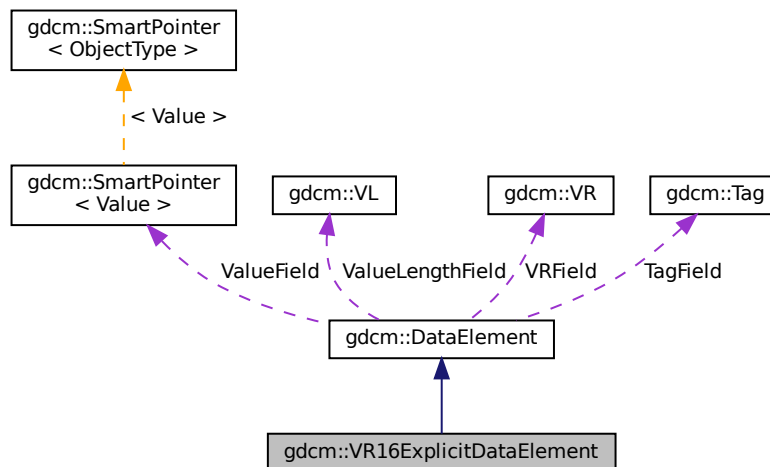
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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)`

## Additional Inherited Members

### 10.370.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

This class support 16 bits when finding an unkown [VR](#): For instance: Siemens\_CT\_Sensation64\_has\_VR\_RT.dcm

### 10.370.2 Member Function Documentation

#### 10.370.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength ( ) const
```

#### 10.370.2.2 Read()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::Read (
    std::istream & is )
```

#### 10.370.2.3 ReadPreValue()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (
    std::istream & is )
```

#### 10.370.2.4 ReadValue()

```
template<typename TSwap >
std::istream& gdcm::VR16ExplicitDataElement::ReadValue (
    std::istream & is,
    bool readvalues = true )
```

#### 10.370.2.5 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](#)

### 10.371 gdcm::VRToEncoding< T > Struct Template Reference

```
#include <gdcmVR.h>
```

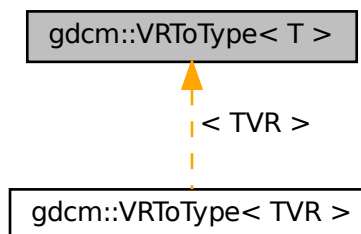
The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

### 10.372 gdcm::VRToType< T > Struct Template Reference

```
#include <gdcmVR.h>
```

Inheritance diagram for gdcm::VRToType< T >:



### 10.372.1 Detailed Description

```
template<int T>
struct gdcm::VRToType< T >
```

Examples:

[DumpGEMSMovieGroup.cxx](#).

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

- [gdcmVR.h](#)

## 10.373 gdcm::VRVLSIZE< T > Class Template Reference

```
#include <gdcmAttribute.h>
```

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

- [gdcmAttribute.h](#)

## 10.374 gdcm::VRVLSIZE< 0 > Class Template Reference

```
#include <gdcmAttribute.h>
```

### Static Public Member Functions

- static uint16\_t [Read](#) (std::istream &\_is)
- static void [Write](#) (std::ostream &os)

### 10.374.1 Member Function Documentation

#### 10.374.1.1 Read()

```
static uint16_t gdcm::VRVLSIZE< 0 >::Read (
    std::istream & _is ) [inline], [static]
```

#### 10.374.1.2 Write()

```
static void gdcm::VRVLSize< 0 >::Write (
    std::ostream & os ) [inline], [static]
```

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

- [gdcmAttribute.h](#)

### 10.375 gdcm::VRVLSize< 1 > Class Template Reference

```
#include <gdcmAttribute.h>
```

#### Static Public Member Functions

- static uint32\_t [Read](#) (std::istream &\_is)
- static void [Write](#) (std::ostream &os)

#### 10.375.1 Member Function Documentation

##### 10.375.1.1 Read()

```
static uint32_t gdcm::VRVLSize< 1 >::Read (
    std::istream & _is ) [inline], [static]
```

##### 10.375.1.2 Write()

```
static void gdcm::VRVLSize< 1 >::Write (
    std::ostream & os ) [inline], [static]
```

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

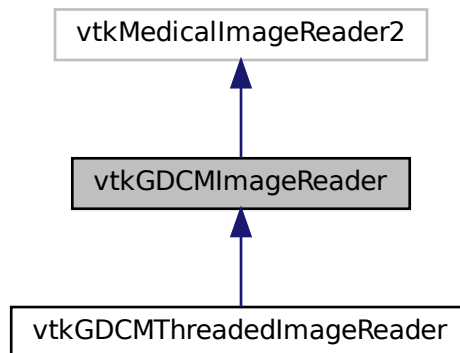
- [gdcmAttribute.h](#)



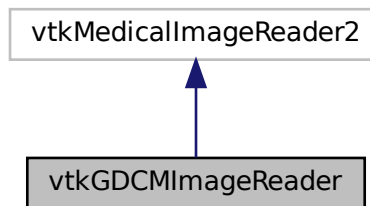
## 10.376 vtkGDCMImageReader Class Reference

```
#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 \* [GetIcnImage](#) ()
- 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](#) (LoadOverlays, int)
- [vtkBooleanMacro](#) (LoadIconImage, int)
- [vtkBooleanMacro](#) (LossyFlag, int)
- [vtkBooleanMacro](#) (ApplyLookupTable, int)
- int [vtkBooleanMacro](#) (ApplyYBRToRGB, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (LoadIconImage, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (NumberOfIconImages, int)
- [vtkGetMacro](#) (ApplyLookupTable, int)
- [vtkGetMacro](#) (ApplyYBRToRGB, int) [vtkSetMacro](#)(ApplyYBRToRGB
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (Curve, vtkPolyData)
- [vtkGetVector3Macro](#) (ImagePositionPatient, double)
- [vtkGetVector6Macro](#) (ImageOrientationPatient, double)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (LoadIconImage, int)
- [vtkSetMacro](#) (LossyFlag, int)
- [vtkSetMacro](#) (ApplyLookupTable, 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](#) (FilePrefix)
- [vtkGetStringMacro](#) (FilePattern)
- [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](#)

### 10.376.1 Detailed Description

#### Examples:

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmttexture.cxx](#), [gdcmvolume.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [MagnifyFile.cxx](#), [MetaImageMD5Activiz.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), [offscreenimage.cxx](#), [ReadSeriesIntoVTK.java](#), [RefCounting.cs](#), and [reslicesphere.cxx](#).

### 10.376.2 Constructor & Destructor Documentation

#### 10.376.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader ( ) [protected]
```

#### Examples:

[HelloActiviz2.cs](#).

### 10.376.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~~vtkGDCMImageReader ( ) [protected]
```

## 10.376.3 Member Function Documentation

### 10.376.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
    const char * fname ) [virtual]
```

#### Examples:

[MetaImageMD5Activiz.cs](#).

### 10.376.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.376.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ( ) [protected]
```

### 10.376.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
    const gdcm::ImageReader & reader ) [protected]
```

### 10.376.3.5 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]
```

### 10.376.3.6 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]
```

### 10.376.3.7 GetIconImage()

```
vtkImageData* vtkGDCMImageReader::GetIconImage ( )
```

### 10.376.3.8 GetOverlay()

```
vtkImageData* vtkGDCMImageReader::GetOverlay (
    int i )
```

### 10.376.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

### 10.376.3.10 New()

```
static vtkGDCMImageReader* vtkGDCMImageReader::New ( ) [static]
```

#### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmtexture.cxx](#), [gdcmvolume.cxx](#), [HelloActiviz.cs](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [HelloVTKWorld.cs](#), [MagnifyFile.cxx](#), [MetalImageMD5Activiz.cs](#), [offscreenimage.cxx](#), [RefCounting.cs](#), and [reslicesphere.cxx](#).

**10.376.3.11 PrintSelf()**

```
virtual void vtkGDCMImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented in [vtkGDCMThreadedImageReader](#).

**10.376.3.12 RequestDataCompat()**

```
int vtkGDCMImageReader::RequestDataCompat ( ) [protected]
```

**10.376.3.13 RequestInformationCompat()**

```
int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]
```

**10.376.3.14 SetCurve()**

```
virtual void vtkGDCMImageReader::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.376.3.15 SetFileNames()**

```
virtual void vtkGDCMImageReader::SetFileNames (
    vtkStringArray * ) [virtual]
```

**Examples:**

[gdcmmorphoplanes.cxx](#), [HelloActiviz3.cs](#), [HelloActiviz4.cs](#), [HelloActiviz5.cs](#), [MIPViewer.java](#), [MPRViewer.java](#), [MPRViewer2.java](#), and [ReadSeriesIntoVTK.java](#).

**10.376.3.16 SetFilePattern()**

```
void vtkGDCMImageReader::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.376.3.17 SetFilePrefix()**

```
void vtkGDCMImageReader::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.376.3.18 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.376.3.19 vtkBooleanMacro()** [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.376.3.20 vtkBooleanMacro()** [2/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.376.3.21 vtkBooleanMacro()** [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.376.3.22 vtkBooleanMacro()** [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

**10.376.3.23 vtkBooleanMacro()** [5/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
    ApplyYBRToRGB ,
    int )
```

**10.376.3.24 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.376.3.25 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.376.3.26 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.376.3.27 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.376.3.28 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
    NumberOfIconImages ,
    int )
```



**10.376.3.29 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

**10.376.3.30 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.376.3.31 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.376.3.32 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.376.3.33 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Shift ,
    double )
```

**10.376.3.34 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
    Scale ,
    double )
```

**10.376.3.35 vtkGetObjectMacro()** [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.376.3.36 vtkGetObjectMacro()** [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.376.3.37 vtkGetObjectMacro()** [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.376.3.38 vtkGetObjectMacro()** [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.376.3.39 vtkGetStringMacro()** [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.376.3.40 vtkGetStringMacro()** [2/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.376.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.376.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.376.3.43 vtkSetMacro()** [1/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.376.3.44 vtkSetMacro()** [2/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.376.3.45 vtkSetMacro()** [3/4]

```
vtkGDCMImageReader::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.376.3.46 vtkSetMacro()** [4/4]

```
vtkGDCMImageReader::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.376.3.47 vtkSetVector6Macro()**

```
vtkGDCMImageReader::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

**10.376.3.48 vtkTypeMacro()**

```
vtkGDCMImageReader::vtkTypeMacro (
    vtkGDCMImageReader ,
    vtkMedicalImageReader2 )
```

**10.376.4 Member Data Documentation****10.376.4.1 ApplyInverseVideo**

```
int vtkGDCMImageReader::ApplyInverseVideo [protected]
```

**10.376.4.2 ApplyLookupTable**

```
int vtkGDCMImageReader::ApplyLookupTable [protected]
```

**10.376.4.3 ApplyPlanarConfiguration**

```
int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]
```

**10.376.4.4 ApplyShiftScale**

```
int vtkGDCMImageReader::ApplyShiftScale [protected]
```

#### 10.376.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB [protected]
```

#### 10.376.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve [protected]
```

#### 10.376.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]
```

#### 10.376.4.8 FileNames

```
vtkStringArray* vtkGDCMImageReader::FileNames [protected]
```

#### 10.376.4.9 ForceRescale

```
int vtkGDCMImageReader::ForceRescale [protected]
```

#### 10.376.4.10 IconDataScalarType

```
int vtkGDCMImageReader::IconDataScalarType [protected]
```

#### 10.376.4.11 IconImageDataExtent

```
int vtkGDCMImageReader::IconImageDataExtent[6] [protected]
```

**10.376.4.12 IconNumberOfScalarComponents**

```
int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]
```

**10.376.4.13 ImageFormat**

```
int vtkGDCMImageReader::ImageFormat [protected]
```

**10.376.4.14 ImageOrientationPatient**

```
double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]
```

**10.376.4.15 ImagePositionPatient**

```
double vtkGDCMImageReader::ImagePositionPatient[3] [protected]
```

**10.376.4.16 LoadIconImage**

```
int vtkGDCMImageReader::LoadIconImage [protected]
```

**10.376.4.17 LoadOverlays**

```
int vtkGDCMImageReader::LoadOverlays [protected]
```

**10.376.4.18 LossyFlag**

```
int vtkGDCMImageReader::LossyFlag [protected]
```

#### 10.376.4.19 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]
```

#### 10.376.4.20 NumberOfIconImages

```
int vtkGDCMImageReader::NumberOfIconImages [protected]
```

#### 10.376.4.21 NumberOfOverlays

```
int vtkGDCMImageReader::NumberOfOverlays [protected]
```

#### 10.376.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration [protected]
```

#### 10.376.4.23 Scale

```
double vtkGDCMImageReader::Scale [protected]
```

#### 10.376.4.24 Shift

```
double vtkGDCMImageReader::Shift [protected]
```

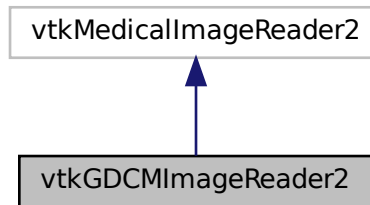
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

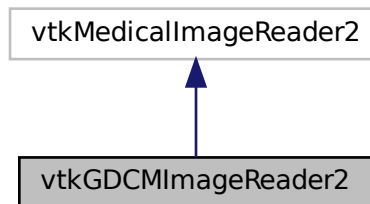
## 10.377 vtkGDCMImageReader2 Class Reference

```
#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](#) ([LoadOverlays](#), int)
- [vtkBooleanMacro](#) ([LoadIconImage](#), int)
- [vtkBooleanMacro](#) ([LossyFlag](#), int)
- [vtkBooleanMacro](#) ([ApplyLookupTable](#), int)
- [int vtkBooleanMacro](#) ([ApplyYBRToRGB](#), int)
- [vtkGetMacro](#) ([LoadOverlays](#), int)
- [vtkGetMacro](#) ([LoadIconImage](#), int)
- [vtkGetMacro](#) ([LossyFlag](#), int)
- [vtkGetMacro](#) ([NumberOfOverlays](#), int)
- [vtkGetMacro](#) ([NumberOfIconImages](#), int)
- [vtkGetMacro](#) ([ApplyLookupTable](#), int)
- [vtkGetMacro](#) ([ApplyYBRToRGB](#), int) [vtkSetMacro](#)([ApplyYBRToRGB](#)
- [vtkGetMacro](#) ([ImageFormat](#), int)
- [vtkGetMacro](#) ([PlanarConfiguration](#), int)
- [vtkGetMacro](#) ([Shift](#), double)
- [vtkGetMacro](#) ([Scale](#), double)
- [vtkGetObjectMacro](#) ([DirectionCosines](#), [vtkMatrix4x4](#))
- [vtkGetObjectMacro](#) ([Curve](#), [vtkPolyData](#))
- [vtkGetVector3Macro](#) ([ImagePositionPatient](#), double)
- [vtkGetVector6Macro](#) ([ImageOrientationPatient](#), double)
- [vtkSetMacro](#) ([LoadOverlays](#), int)
- [vtkSetMacro](#) ([LoadIconImage](#), int)
- [vtkSetMacro](#) ([LossyFlag](#), int)
- [vtkSetMacro](#) ([ApplyLookupTable](#), 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](#) \*output←  
Vector)
- int [RequestData](#) ([vtkInformation](#) \*request, [vtkInformationVector](#) \*\*inputVector, [vtkInformationVector](#) \*output←  
Vector)
- int [RequestDataCompat](#) ()
- int [RequestInformation](#) ([vtkInformation](#) \*request, [vtkInformationVector](#) \*\*inputVector, [vtkInformationVector](#) \*outputVector)
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char \*)
- void [SetFilePrefix](#) (const char \*)
- [vtkGetStringMacro](#) ([FilePrefix](#))
- [vtkGetStringMacro](#) ([FilePattern](#))
- [vtkSetVector6Macro](#) ([ImageOrientationPatient](#), double)

## Protected Attributes

- int [ApplyInverseVideo](#)
- int [ApplyLookupTable](#)
- int [ApplyPlanarConfiguration](#)
- int [ApplyShiftScale](#)
- int [ApplyYBRToRGB](#)
- 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](#)

### 10.377.1 Detailed Description

Examples:

[Compute3DSpacing.cxx](#).

### 10.377.2 Constructor & Destructor Documentation

#### 10.377.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ( ) [protected]
```

#### 10.377.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~~vtkGDCMImageReader2 ( ) [protected]
```

### 10.377.3 Member Function Documentation

#### 10.377.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
    const char * fname ) [virtual]
```

#### 10.377.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
    const gdcm::ImageReader & reader ) [protected]
```

#### 10.377.3.3 GetDescriptiveName()

```
virtual const char* vtkGDCMImageReader2::GetDescriptiveName ( ) [inline], [virtual]
```

#### 10.377.3.4 GetFileExtensions()

```
virtual const char* vtkGDCMImageReader2::GetFileExtensions ( ) [inline], [virtual]
```

#### 10.377.3.5 GetIconImage()

```
vtkImageData* vtkGDCMImageReader2::GetIconImage ( )
```

#### 10.377.3.6 GetIconImagePort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )
```

### 10.377.3.7 GetOverlay()

```
vtkImageData* vtkGDCMImageReader2::GetOverlay (
    int i )
```

### 10.377.3.8 GetOverlayPort()

```
vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (
    int index )
```

### 10.377.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
    const char * filename,
    char * pointer,
    unsigned long & outlen ) [protected]
```

### 10.377.3.10 New()

```
static vtkGDCMImageReader2* vtkGDCMImageReader2::New ( ) [static]
```

#### Examples:

[Compute3DSpacing.cxx](#).

### 10.377.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

**10.377.3.12 ProcessRequest()**

```
int vtkGDCMImageReader2::ProcessRequest (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.377.3.13 RequestData()**

```
int vtkGDCMImageReader2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.377.3.14 RequestDataCompat()**

```
int vtkGDCMImageReader2::RequestDataCompat ( ) [protected]
```

**10.377.3.15 RequestInformation()**

```
int vtkGDCMImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.377.3.16 RequestInformationCompat()**

```
int vtkGDCMImageReader2::RequestInformationCompat ( ) [protected]
```

**10.377.3.17 SetCurve()**

```
virtual void vtkGDCMImageReader2::SetCurve (
    vtkPolyData * pd ) [virtual]
```

**10.377.3.18 SetFilePattern()**

```
void vtkGDCMImageReader2::SetFilePattern (
    const char * ) [inline], [protected]
```

**10.377.3.19 SetFilePrefix()**

```
void vtkGDCMImageReader2::SetFilePrefix (
    const char * ) [inline], [protected]
```

**10.377.3.20 SetMedicalImageProperties()**

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

**10.377.3.21 vtkBooleanMacro()** [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

**10.377.3.22 vtkBooleanMacro()** [2/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LoadIconImage ,
    int )
```

**10.377.3.23 vtkBooleanMacro()** [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    LossyFlag ,
    int )
```

**10.377.3.24 vtkBooleanMacro()** [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyLookupTable ,
    int )
```

**10.377.3.25 vtkBooleanMacro()** [5/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (
    ApplyYBRTToRGB ,
    int )
```

**10.377.3.26 vtkGetMacro()** [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.377.3.27 vtkGetMacro()** [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LoadIconImage ,
    int )
```

**10.377.3.28 vtkGetMacro()** [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    LossyFlag ,
    int )
```

**10.377.3.29 vtkGetMacro()** [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.377.3.30 vtkGetMacro()** [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    NumberOfIconImages ,
    int )
```

**10.377.3.31 vtkGetMacro()** [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyLookupTable ,
    int )
```

**10.377.3.32 vtkGetMacro()** [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ApplyYBRToRGB ,
    int )
```

**10.377.3.33 vtkGetMacro()** [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.377.3.34 vtkGetMacro()** [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.377.3.35 vtkGetMacro()** [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Shift ,
    double )
```



**10.377.3.36 vtkGetMacro()** [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.377.3.37 vtkGetObjectMacro()** [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.377.3.38 vtkGetObjectMacro()** [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
    Curve ,
    vtkPolyData )
```

**10.377.3.39 vtkGetStringMacro()** [1/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePrefix ) [protected]
```

**10.377.3.40 vtkGetStringMacro()** [2/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
    FilePattern ) [protected]
```

**10.377.3.41 vtkGetVector3Macro()**

```
vtkGDCMImageReader2::vtkGetVector3Macro (
    ImagePositionPatient ,
    double )
```

**10.377.3.42 vtkGetVector6Macro()**

```
vtkGDCMImageReader2::vtkGetVector6Macro (
    ImageOrientationPatient ,
    double )
```

**10.377.3.43 vtkSetMacro()** [1/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.377.3.44 vtkSetMacro()** [2/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LoadIconImage ,
    int )
```

**10.377.3.45 vtkSetMacro()** [3/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.377.3.46 vtkSetMacro()** [4/4]

```
vtkGDCMImageReader2::vtkSetMacro (
    ApplyLookupTable ,
    int )
```

**10.377.3.47 vtkSetVector6Macro()**

```
vtkGDCMImageReader2::vtkSetVector6Macro (
    ImageOrientationPatient ,
    double ) [protected]
```

### 10.377.3.48 vtkTypeMacro()

```
vtkGDCMImageReader2::vtkTypeMacro (
    vtkGDCMImageReader2 ,
    vtkMedicalImageReader2 )
```

## 10.377.4 Member Data Documentation

### 10.377.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader2::ApplyInverseVideo [protected]
```

### 10.377.4.2 ApplyLookupTable

```
int vtkGDCMImageReader2::ApplyLookupTable [protected]
```

### 10.377.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]
```

### 10.377.4.4 ApplyShiftScale

```
int vtkGDCMImageReader2::ApplyShiftScale [protected]
```

### 10.377.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader2::ApplyYBRToRGB [protected]
```

**10.377.4.6 Curve**

```
vtkPolyData* vtkGDCMImageReader2::Curve [protected]
```

**10.377.4.7 DirectionCosines**

```
vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines [protected]
```

**10.377.4.8 ForceRescale**

```
int vtkGDCMImageReader2::ForceRescale [protected]
```

**10.377.4.9 IconDataScalarType**

```
int vtkGDCMImageReader2::IconDataScalarType [protected]
```

**10.377.4.10 IconImageDataExtent**

```
int vtkGDCMImageReader2::IconImageDataExtent[6] [protected]
```

**10.377.4.11 IconNumberOfScalarComponents**

```
int vtkGDCMImageReader2::IconNumberOfScalarComponents [protected]
```

**10.377.4.12 ImageFormat**

```
int vtkGDCMImageReader2::ImageFormat [protected]
```

**10.377.4.13 ImageOrientationPatient**

```
double vtkGDCMImageReader2::ImageOrientationPatient[6] [protected]
```

**10.377.4.14 ImagePositionPatient**

```
double vtkGDCMImageReader2::ImagePositionPatient[3] [protected]
```

**10.377.4.15 LoadIconImage**

```
int vtkGDCMImageReader2::LoadIconImage [protected]
```

**10.377.4.16 LoadOverlays**

```
int vtkGDCMImageReader2::LoadOverlays [protected]
```

**10.377.4.17 LossyFlag**

```
int vtkGDCMImageReader2::LossyFlag [protected]
```

**10.377.4.18 NumberOfIconImages**

```
int vtkGDCMImageReader2::NumberOfIconImages [protected]
```

**10.377.4.19 NumberOfOverlays**

```
int vtkGDCMImageReader2::NumberOfOverlays [protected]
```

### 10.377.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration [protected]
```

### 10.377.4.21 Scale

```
double vtkGDCMImageReader2::Scale [protected]
```

### 10.377.4.22 Shift

```
double vtkGDCMImageReader2::Shift [protected]
```

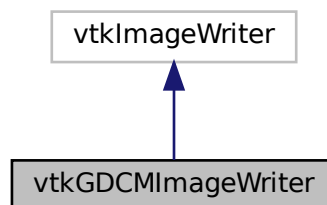
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

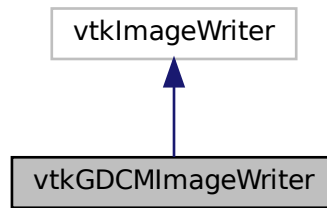
## 10.378 vtkGDCMImageWriter Class Reference

```
#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](#) (LossyFlag, int)
- [vtkBooleanMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (LossyFlag, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (ImageFormat, int)
- [vtkGetMacro](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (PlanarConfiguration, int)
- [vtkGetMacro](#) (CompressionType, int)
- [vtkGetObjectMacro](#) (MedicalImageProperties, vtkMedicalImageProperties)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetObjectMacro](#) (DirectionCosines, vtkMatrix4x4)
- [vtkGetStringMacro](#) (StudyUID)
- [vtkGetStringMacro](#) (SeriesUID)
- [vtkSetMacro](#) (LossyFlag, int)

- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (ImageFormat, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (PlanarConfiguration, int)
- [vtkSetMacro](#) (CompressionType, int)
- [vtkSetStringMacro](#) (StudyUID)
- [vtkSetStringMacro](#) (SeriesUID)
- [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)

## 10.378.1 Detailed Description

### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmorthoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

## 10.378.2 Member Enumeration Documentation

### 10.378.2.1 CompressionTypes

enum [vtkGDCMImageWriter::CompressionTypes](#)

#### Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	



### 10.378.3 Constructor & Destructor Documentation

#### 10.378.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter ( ) [protected]
```

#### 10.378.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~~vtkGDCMImageWriter ( ) [protected]
```

### 10.378.4 Member Function Documentation

#### 10.378.4.1 GetDescriptiveName()

```
virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]
```

#### 10.378.4.2 GetFileExtensions()

```
virtual const char* vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]
```

#### 10.378.4.3 GetFileName()

```
virtual char* vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]
```

#### 10.378.4.4 New()

```
static vtkGDCMImageWriter* vtkGDCMImageWriter::New ( ) [static]
```

##### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmorthoplanes.cxx](#), [HelloActiviz.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

#### 10.378.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.378.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
    vtkMatrix4x4 * matrix ) [virtual]
```

##### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorthoplanes.cxx](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), and [MagnifyFile.cxx](#).

#### 10.378.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
    const double dircos[6] ) [virtual]
```

#### 10.378.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
    vtkStringArray * ) [virtual]
```

##### Examples:

[ConvertMultiFrameToSingleFrame.cxx](#), and [CreateFakePET.cxx](#).

#### 10.378.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * ) [virtual]
```

##### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorthoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), and [MagnifyFile.cxx](#).

#### 10.378.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    LossyFlag ,
    int )
```

#### 10.378.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

#### 10.378.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    LossyFlag ,
    int )
```

#### 10.378.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Shift ,
    double )
```

**10.378.4.14 vtkGetMacro()** [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    Scale ,
    double )
```

**10.378.4.15 vtkGetMacro()** [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    ImageFormat ,
    int )
```

**10.378.4.16 vtkGetMacro()** [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    FileLowerLeft ,
    int )
```

**10.378.4.17 vtkGetMacro()** [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    PlanarConfiguration ,
    int )
```

**10.378.4.18 vtkGetMacro()** [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
    CompressionType ,
    int )
```

**10.378.4.19 vtkGetObjectMacro()** [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.378.4.20 vtkGetObjectMacro()** [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.378.4.21 vtkGetObjectMacro()** [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
    DirectionCosines ,
    vtkMatrix4x4 )
```

**10.378.4.22 vtkGetStringMacro()** [1/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
    StudyUID )
```

**10.378.4.23 vtkGetStringMacro()** [2/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
    SeriesUID )
```

**10.378.4.24 vtkSetMacro()** [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    LossyFlag ,
    int )
```

**10.378.4.25 vtkSetMacro()** [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    Shift ,
    double )
```

**10.378.4.26 vtkSetMacro()** [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    Scale ,
    double )
```

**10.378.4.27 vtkSetMacro()** [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    ImageFormat ,
    int )
```

**10.378.4.28 vtkSetMacro()** [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.378.4.29 vtkSetMacro()** [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    PlanarConfiguration ,
    int )
```

**10.378.4.30 vtkSetMacro()** [7/7]

```
vtkGDCMImageWriter::vtkSetMacro (
    CompressionType ,
    int )
```

**10.378.4.31 vtkSetStringMacro()** [1/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
    StudyUID )
```

**10.378.4.32 vtkSetStringMacro()** [2/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
    SeriesUID )
```

**10.378.4.33 vtkTypeMacro()**

```
vtkGDCMImageWriter::vtkTypeMacro (
    vtkGDCMImageWriter ,
    vtkImageWriter )
```

**10.378.4.34 Write()**

```
virtual void vtkGDCMImageWriter::Write ( ) [virtual]
```

**Examples:**

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [CreateFakePET.cxx](#), [CreateFakeRTDOSE.cxx](#), [gdcmothoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), and [MagnifyFile.cxx](#).

**10.378.4.35 WriteGDCMData()**

```
int vtkGDCMImageWriter::WriteGDCMData (
    vtkImageData * data,
    int timeStep ) [protected]
```

**10.378.4.36 WriteSlice()**

```
void vtkGDCMImageWriter::WriteSlice (
    vtkImageData * data ) [protected]
```

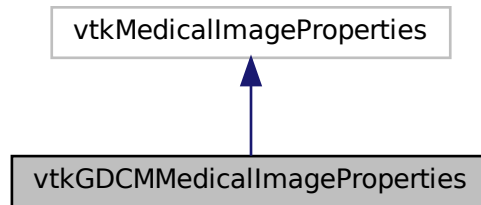
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

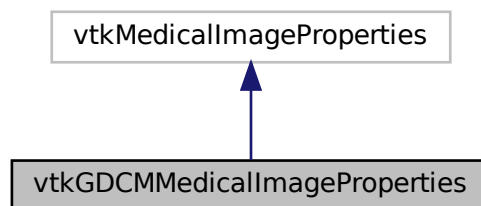
## 10.379 vtkGDCMMedicalImageProperties Class Reference

```
#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](#)

## 10.379.1 Constructor & Destructor Documentation

### 10.379.1.1 vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]
```

### 10.379.1.2 ~vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ( ) [protected]
```

## 10.379.2 Member Function Documentation

### 10.379.2.1 Clear()

```
virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]
```

### 10.379.2.2 GetFile()

```
gdcmm::File const& vtkGDCMMedicalImageProperties::GetFile (
    unsigned int t ) [protected]
```

#### 10.379.2.3 New()

```
static vtkGDCMMedicalImageProperties\* vtkGDCMMedicalImageProperties::New ( ) [static]
```

#### 10.379.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.379.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
    gdcmm::File const & f ) [protected]
```

#### 10.379.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
    vtkGDCMMedicalImageProperties ,
    vtkMedicalImageProperties )
```

### 10.379.3 Friends And Related Function Documentation

#### 10.379.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader [friend]
```

#### 10.379.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2 [friend]
```

## 10.379.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter [friend]
```

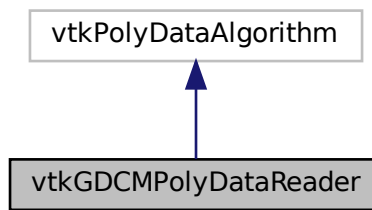
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

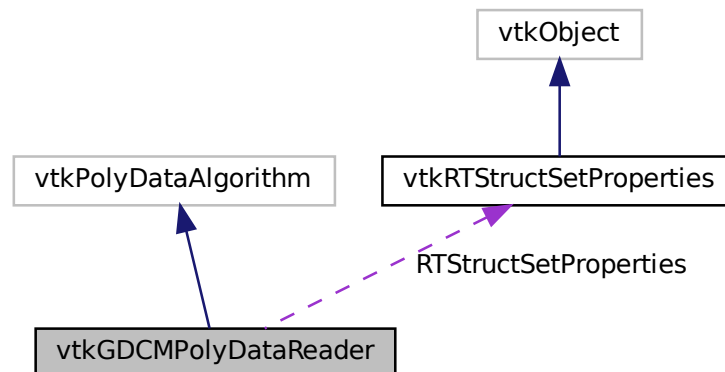
## 10.380 vtkGDCMPolyDataReader Class Reference

```
#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](#)

### 10.380.1 Detailed Description

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.380.2 Constructor & Destructor Documentation

### 10.380.2.1 vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]
```

### 10.380.2.2 ~vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ( ) [protected]
```

## 10.380.3 Member Function Documentation

### 10.380.3.1 FillMedicalImageInformation()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
    const gdcmm::Reader & reader ) [protected]
```

### 10.380.3.2 New()

```
static vtkGDCMPolyDataReader\* vtkGDCMPolyDataReader::New ( ) [static]
```

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.380.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

#### 10.380.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected]
```

#### 10.380.3.5 RequestData\_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
    gdcM::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.380.3.6 RequestData\_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
    gdcM::Reader const & reader,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.380.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
    vtkInformation * vtkNotUsedrequest,
    vtkInformationVector ** vtkNotUsedinputVector,
    vtkInformationVector * outputVector ) [protected]
```

#### 10.380.3.8 RequestInformation\_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
    gdcM::Reader const & reader ) [protected]
```

#### 10.380.3.9 RequestInformation\_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
    gdcM::Reader const & reader ) [protected]
```

**10.380.3.10 vtkGetObjectMacro()** [1/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    MedicalImageProperties ,
    vtkMedicalImageProperties )
```

**10.380.3.11 vtkGetObjectMacro()** [2/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
    RTStructSetProperties ,
    vtkRTStructSetProperties )
```

**10.380.3.12 vtkGetStringMacro()**

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
    FileName )
```

**10.380.3.13 vtkSetStringMacro()**

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
    FileName )
```

**10.380.3.14 vtkTypeMacro()**

```
vtkGDCMPolyDataReader::vtkTypeMacro (
    vtkGDCMPolyDataReader ,
    vtkPolyDataAlgorithm )
```

**10.380.4 Member Data Documentation****10.380.4.1 FileName**

```
char* vtkGDCMPolyDataReader::FileName [protected]
```

#### 10.380.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]
```

#### 10.380.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]
```

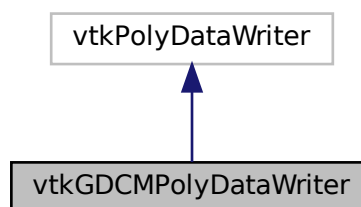
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

### 10.381 vtkGDCMPolyDataWriter Class Reference

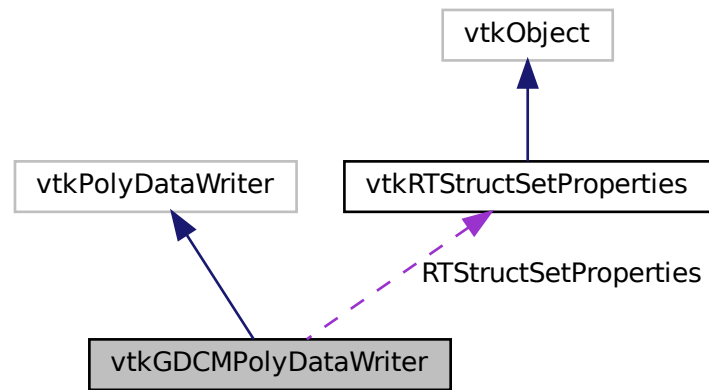
```
#include <vtkGDCMPolyDataWriter.h>
```

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](#) (gdcm::File &file, int num)
- void [WriteRTSTRUCTInfo](#) (gdcm::File &file)

### Protected Attributes

- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) \* [RTStructSetProperties](#)

### 10.381.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.2 Constructor & Destructor Documentation

#### 10.381.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]
```

#### 10.381.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ( ) [protected]
```

### 10.381.3 Member Function Documentation

#### 10.381.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
    vtkStdString inDirectory,
    vtkStdString inStructLabel,
    vtkStdString inStructName,
    vtkStringArray * inROINames,
    vtkStringArray * inROIAlgorithmName,
    vtkStringArray * inROIType )
```

Examples:

[GenerateRTSTRUCT.cxx](#).

### 10.381.3.2 New()

```
static vtkGDCMPolyDataWriter\* vtkGDCMPolyDataWriter::New ( ) [static]
```

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

### 10.381.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
    vtkMedicalImageProperties * pd ) [virtual]
```

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 10.381.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
    int n )
```

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

#### 10.381.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
    vtkRTStructSetProperties * pd ) [virtual]
```

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

#### 10.381.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
    vtkGDCMPolyDataWriter ,
    vtkPolyDataWriter )
```

#### 10.381.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData ( ) [protected]
```

#### 10.381.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
    gdcM::File & file,
    int num ) [protected]
```

#### 10.381.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
    gdcM::File & file ) [protected]
```

### 10.381.4 Member Data Documentation

#### 10.381.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]
```

#### 10.381.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]
```

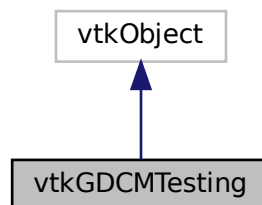
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

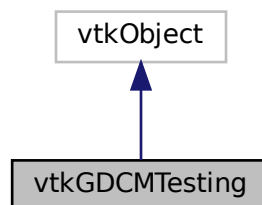
## 10.382 vtkGDCMTesting Class Reference

```
#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](#) ()

### 10.382.1 Detailed Description

Examples:

[HelloActiviz5.cs](#), [HelloVTKWorld2.cs](#), [MetaImageMD5Activiz.cs](#), [ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

### 10.382.2 Member Typedef Documentation

#### 10.382.2.1 MD5MetaImagesType

```
typedef const char* const (* vtkGDCMTesting::MD5MetaImagesType) [3]
```

### 10.382.3 Constructor & Destructor Documentation

### 10.382.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ( ) [protected]
```

### 10.382.3.2 ~vtkGDCMTesting()

```
vtkGDCMTesting::~~vtkGDCMTesting ( ) [protected]
```

## 10.382.4 Member Function Documentation

### 10.382.4.1 GetGDCMDataRoot()

```
static const char* vtkGDCMTesting::GetGDCMDataRoot ( ) [static]
```

#### Examples:

[HelloActiviz5.cs](#), and [ReadSeriesIntoVTK.java](#).

### 10.382.4.2 GetMD5MetaImage()

```
static const char* const* vtkGDCMTesting::GetMD5MetaImage (
    unsigned int file ) [static]
```

### 10.382.4.3 GetMHDMD5FromFile()

```
static const char* vtkGDCMTesting::GetMHDMD5FromFile (
    const char * filepath ) [static]
```

#### Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.382.4.4 GetNumberOfMD5MetaImages()

```
static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]
```

#### 10.382.4.5 GetRAWMD5FromFile()

```
static const char* vtkGDCMTesting::GetRAWMD5FromFile (
    const char * filepath ) [static]
```

##### Examples:

[MetaImageMD5Activiz.cs](#).

#### 10.382.4.6 GetVTKDataRoot()

```
static const char* vtkGDCMTesting::GetVTKDataRoot ( ) [static]
```

##### Examples:

[HelloActiviz5.cs](#), and [HelloVTKWorld2.cs](#).

#### 10.382.4.7 New()

```
static vtkGDCMTesting\* vtkGDCMTesting::New ( ) [static]
```

##### Examples:

[RefCounting.cs](#).

#### 10.382.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
    ostream & os,
    vtkIndent indent )
```



## 10.382.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (
    vtkGDCMTesting ,
    vtkObject )
```

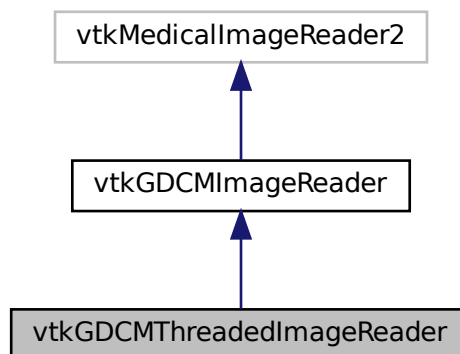
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

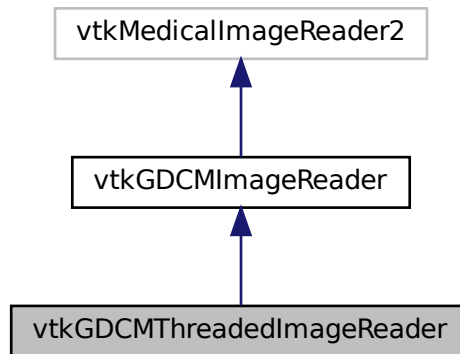
## 10.383 vtkGDCMThreadedImageReader Class Reference

```
#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](#) (Shift, double)
- [vtkSetMacro](#) (Scale, double)
- [vtkSetMacro](#) (UseShiftScale, int)
- [vtkTypeMacro](#) (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

### Static Public Member Functions

- static [vtkGDCMThreadedImageReader \\* New](#) ()

### Protected Member Functions

- [vtkGDCMThreadedImageReader](#) ()
- [~vtkGDCMThreadedImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject \*out)
- void [ExecuteInformation](#) ()
- void [ReadFiles](#) (unsigned int nfiles, const char \*filenames[])
- void [RequestDataCompat](#) ()

### Additional Inherited Members

#### 10.383.1 Constructor & Destructor Documentation

### 10.383.1.1 vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]
```

### 10.383.1.2 ~vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader ( ) [protected]
```

## 10.383.2 Member Function Documentation

### 10.383.2.1 ExecuteData()

```
void vtkGDCMThreadedImageReader::ExecuteData (
    vtkDataObject * out ) [protected]
```

### 10.383.2.2 ExecuteInformation()

```
void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]
```

### 10.383.2.3 New()

```
static vtkGDCMThreadedImageReader\* vtkGDCMThreadedImageReader::New ( ) [static]
```

### 10.383.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

Reimplemented from [vtkGDCMImageReader](#).

**10.383.2.5 ReadFiles()**

```
void vtkGDCMThreadedImageReader::ReadFiles (
    unsigned int nfiles,
    const char * filenames[] ) [protected]
```

**10.383.2.6 RequestDataCompat()**

```
void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]
```

**10.383.2.7 vtkBooleanMacro()**

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

**10.383.2.8 vtkGetMacro()**

```
vtkGDCMThreadedImageReader::vtkGetMacro (
    UseShiftScale ,
    int )
```

**10.383.2.9 vtkSetMacro()** [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Shift ,
    double )
```

**10.383.2.10 vtkSetMacro()** [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    Scale ,
    double )
```

## 10.383.2.11 vtkSetMacro() [3/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
    UseShiftScale ,
    int )
```

## 10.383.2.12 vtkTypeMacro()

```
vtkGDCMThreadedImageReader::vtkTypeMacro (
    vtkGDCMThreadedImageReader ,
    vtkGDCMImageReader )
```

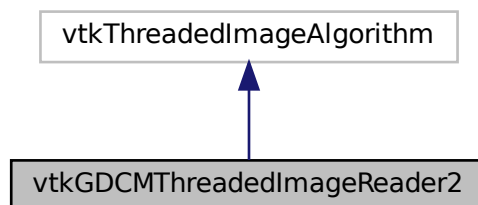
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

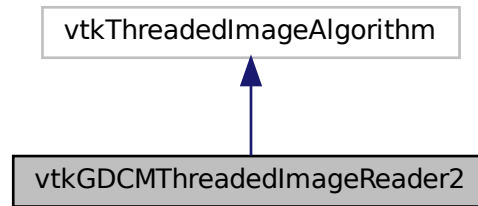
## 10.384 vtkGDCMThreadedImageReader2 Class Reference

```
#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](#) (FileLowerLeft, int)
- [vtkGetMacro](#) (NumberOfOverlays, int)
- [vtkGetMacro](#) (DataScalarType, int)
- [vtkGetMacro](#) (NumberOfScalarComponents, int)
- [vtkGetMacro](#) (LoadOverlays, int)
- [vtkGetMacro](#) (Shift, double)
- [vtkGetMacro](#) (Scale, double)
- [vtkGetMacro](#) (UseShiftScale, int)
- [vtkGetObjectMacro](#) (FileNames, vtkStringArray)
- [vtkGetVector3Macro](#) (DataOrigin, double)
- [vtkGetVector3Macro](#) (DataSpacing, double)
- [vtkGetVector6Macro](#) (DataExtent, int)
- [vtkSetMacro](#) (FileLowerLeft, int)
- [vtkSetMacro](#) (DataScalarType, int)
- [vtkSetMacro](#) (NumberOfScalarComponents, int)
- [vtkSetMacro](#) (LoadOverlays, int)
- [vtkSetMacro](#) (Shift, double)
- [vtkSetMacro](#) (Scale, 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)

### 10.384.1 Constructor & Destructor Documentation

#### 10.384.1.1 [vtkGDCMThreadedImageReader2\(\)](#)

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]
```

#### 10.384.1.2 [~vtkGDCMThreadedImageReader2\(\)](#)

```
vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2 ( ) [protected]
```

### 10.384.2 Member Function Documentation

#### 10.384.2.1 [GetFileName\(\)](#)

```
virtual const char* vtkGDCMThreadedImageReader2::GetFileName (
    int i = 0 ) [virtual]
```

#### 10.384.2.2 [New\(\)](#)

```
static vtkGDCMThreadedImageReader2\* vtkGDCMThreadedImageReader2::New ( ) [static]
```

**10.384.2.3 PrintSelf()**

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
    ostream & os,
    vtkIndent indent ) [virtual]
```

**10.384.2.4 RequestInformation()**

```
int vtkGDCMThreadedImageReader2::RequestInformation (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected]
```

**10.384.2.5 SetFileName()**

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
    const char * filename ) [virtual]
```

**10.384.2.6 SetFileNames()**

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
    vtkStringArray * ) [virtual]
```

**10.384.2.7 SplitExtent()**

```
int vtkGDCMThreadedImageReader2::SplitExtent (
    int splitExt[6],
    int startExt[6],
    int num,
    int total )
```



### 10.384.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int outExt[6],
    int id ) [protected]
```

### 10.384.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    FileLowerLeft ,
    int )
```

### 10.384.2.10 vtkBooleanMacro() [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    LoadOverlays ,
    int )
```

### 10.384.2.11 vtkBooleanMacro() [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
    UseShiftScale ,
    int )
```

### 10.384.2.12 vtkGetMacro() [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    FileLowerLeft ,
    int )
```

**10.384.2.13 vtkGetMacro()** [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfOverlays ,
    int )
```

**10.384.2.14 vtkGetMacro()** [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    DataScalarType ,
    int )
```

**10.384.2.15 vtkGetMacro()** [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.384.2.16 vtkGetMacro()** [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    LoadOverlays ,
    int )
```

**10.384.2.17 vtkGetMacro()** [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Shift ,
    double )
```

**10.384.2.18 vtkGetMacro()** [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    Scale ,
    double )
```

**10.384.2.19 vtkGetMacro()** [8/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
    UseShiftScale ,
    int )
```

**10.384.2.20 vtkGetObjectMacro()**

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
    FileNames ,
    vtkStringArray )
```

**10.384.2.21 vtkGetVector3Macro()** [1/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataOrigin ,
    double )
```

**10.384.2.22 vtkGetVector3Macro()** [2/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
    DataSpacing ,
    double )
```

**10.384.2.23 vtkGetVector6Macro()**

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
    DataExtent ,
    int )
```

**10.384.2.24 vtkSetMacro()** [1/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    FileLowerLeft ,
    int )
```

**10.384.2.25 vtkSetMacro()** [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    DataScalarType ,
    int )
```

**10.384.2.26 vtkSetMacro()** [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    NumberOfScalarComponents ,
    int )
```

**10.384.2.27 vtkSetMacro()** [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    LoadOverlays ,
    int )
```

**10.384.2.28 vtkSetMacro()** [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Shift ,
    double )
```

**10.384.2.29 vtkSetMacro()** [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    Scale ,
    double )
```

**10.384.2.30 vtkSetMacro()** [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
    UseShiftScale ,
    int )
```

**10.384.2.31** `vtkSetVector3Macro()` [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataOrigin ,
    double )
```

**10.384.2.32** `vtkSetVector3Macro()` [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
    DataSpacing ,
    double )
```

**10.384.2.33** `vtkSetVector6Macro()`

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
    DataExtent ,
    int )
```

**10.384.2.34** `vtkTypeMacro()`

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
    vtkGDCMThreadedImageReader2 ,
    vtkThreadedImageAlgorithm )
```

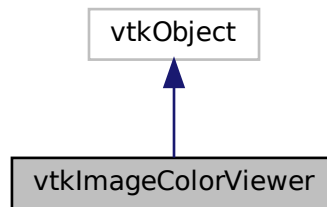
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

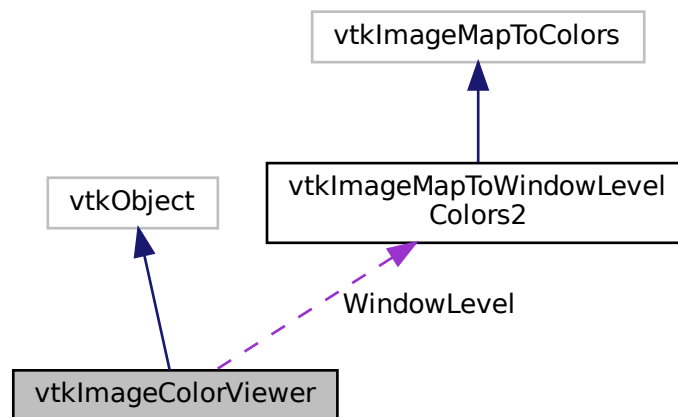
## 10.385 vtkImageColorViewer Class Reference

```
#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 void [GetSliceRange](#) (int range[2])
- virtual void [GetSliceRange](#) (int &min, int &max)
- virtual int \* [GetSliceRange](#) ()
- 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](#) ()
- [VTK\\_LEGACY](#) (int GetWholeZMin())
- [VTK\\_LEGACY](#) (int GetWholeZMax())
- [VTK\\_LEGACY](#) (int GetZSlice())
- [VTK\\_LEGACY](#) (void SetZSlice(int))
- [vtkBooleanMacro](#) (OffScreenRendering, int)
- [vtkGetMacro](#) (SliceOrientation, int)
- [vtkGetMacro](#) (Slice, int)
- [vtkGetObjectMacro](#) (RenderWindow, vtkRenderWindow)
- [vtkGetObjectMacro](#) (Renderer, vtkRenderer)
- [vtkGetObjectMacro](#) (ImageActor, vtkImageActor)
- [vtkGetObjectMacro](#) (WindowLevel, vtkImageMapToWindowLevelColors2)
- [vtkGetObjectMacro](#) (InteractorStyle, vtkInteractorStyleImage)
- [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](#)

## 10.385.1 Detailed Description

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

## 10.385.2 Member Enumeration Documentation

### 10.385.2.1 anonymous enum

anonymous enum



## Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

### 10.385.3 Constructor & Destructor Documentation

#### 10.385.3.1 vtkImageColorViewer()

```
vtkImageColorViewer::vtkImageColorViewer ( ) [protected]
```

#### 10.385.3.2 ~vtkImageColorViewer()

```
vtkImageColorViewer::~~vtkImageColorViewer ( ) [protected]
```

### 10.385.4 Member Function Documentation

#### 10.385.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
    vtkImageData * input ) [virtual]
```

#### 10.385.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

**10.385.4.3 GetColorLevel()**

```
virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]
```

**10.385.4.4 GetColorWindow()**

```
virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]
```

**10.385.4.5 GetInput()**

```
virtual vtkImageData* vtkImageColorViewer::GetInput ( ) [virtual]
```

**10.385.4.6 GetOffScreenRendering()**

```
virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]
```

**10.385.4.7 GetOverlayVisibility()**

```
double vtkImageColorViewer::GetOverlayVisibility ( )
```

**10.385.4.8 GetPosition()**

```
virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]
```

**10.385.4.9 GetSize()**

```
virtual int* vtkImageColorViewer::GetSize ( ) [virtual]
```

**10.385.4.10 GetSliceMax()**

```
virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]
```

**10.385.4.11 GetSliceMin()**

```
virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]
```

**10.385.4.12 GetSliceRange()** [1/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
    int range[2] ) [inline], [virtual]
```

**10.385.4.13 GetSliceRange()** [2/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
    int & min,
    int & max ) [virtual]
```

**10.385.4.14 GetSliceRange()** [3/3]

```
virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]
```

**10.385.4.15 GetWindowName()**

```
virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]
```

**10.385.4.16 InstallPipeline()**

```
virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]
```

**10.385.4.17 New()**

```
static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]
```

**Examples:**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.385.4.18 PrintSelf()**

```
void vtkImageColorViewer::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

**10.385.4.19 Render()**

```
virtual void vtkImageColorViewer::Render (
    void ) [virtual]
```

**Examples:**

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

**10.385.4.20 SetColorLevel()**

```
virtual void vtkImageColorViewer::SetColorLevel (
    double s ) [virtual]
```

**10.385.4.21 SetColorWindow()**

```
virtual void vtkImageColorViewer::SetColorWindow (
    double s ) [virtual]
```

#### 10.385.4.22 SetDisplayId()

```
virtual void vtkImageColorViewer::SetDisplayId (
    void * a ) [virtual]
```

#### 10.385.4.23 SetInput()

```
virtual void vtkImageColorViewer::SetInput (
    vtkImageData * in ) [virtual]
```

#### Examples:

[gdcmrptionplan.cxx](#), and [gdcmrtpplan.cxx](#).

#### 10.385.4.24 SetInputConnection()

```
virtual void vtkImageColorViewer::SetInputConnection (
    vtkAlgorithmOutput * input ) [virtual]
```

#### 10.385.4.25 SetOffScreenRendering()

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
    int ) [virtual]
```

#### 10.385.4.26 SetOverlayVisibility()

```
void vtkImageColorViewer::SetOverlayVisibility (
    double vis )
```

#### 10.385.4.27 SetParentId()

```
virtual void vtkImageColorViewer::SetParentId (
    void * a ) [virtual]
```

**10.385.4.28 SetPosition()** [1/2]

```
virtual void vtkImageColorViewer::SetPosition (
    int a,
    int b ) [virtual]
```

**10.385.4.29 SetPosition()** [2/2]

```
virtual void vtkImageColorViewer::SetPosition (
    int a[2] ) [inline], [virtual]
```

References SetPosition().

Referenced by SetPosition().

**10.385.4.30 SetRenderer()**

```
virtual void vtkImageColorViewer::SetRenderer (
    vtkRenderer * arg ) [virtual]
```

**10.385.4.31 SetRenderWindow()**

```
virtual void vtkImageColorViewer::SetRenderWindow (
    vtkRenderWindow * arg ) [virtual]
```

**10.385.4.32 SetSize()** [1/2]

```
virtual void vtkImageColorViewer::SetSize (
    int a,
    int b ) [virtual]
```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

**10.385.4.33 SetSize()** [2/2]

```
virtual void vtkImageColorViewer::SetSize (
    int a[2] ) [inline], [virtual]
```

References SetSize().

Referenced by SetSize().

**10.385.4.34 SetSlice()**

```
virtual void vtkImageColorViewer::SetSlice (
    int s ) [virtual]
```

**10.385.4.35 SetSliceOrientation()**

```
virtual void vtkImageColorViewer::SetSliceOrientation (
    int orientation ) [virtual]
```

**10.385.4.36 SetSliceOrientationToXY()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_XY.

**10.385.4.37 SetSliceOrientationToXZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_XZ.

**10.385.4.38 SetSliceOrientationToYZ()**

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline], [virtual]
```

References SLICE\_ORIENTATION\_YZ.

**10.385.4.39 SetupInteractor()**

```
virtual void vtkImageColorViewer::SetupInteractor (
    vtkRenderWindowInteractor * ) [virtual]
```

Examples:

[gdcmrptionplan.cxx](#), and [gdcmrtpian.cxx](#).

**10.385.4.40 SetWindowId()**

```
virtual void vtkImageColorViewer::SetWindowId (
    void * a ) [virtual]
```

**10.385.4.41 UnInstallPipeline()**

```
virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected], [virtual]
```

**10.385.4.42 UpdateDisplayExtent()**

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]
```

**10.385.4.43 UpdateOrientation()**

```
virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected], [virtual]
```

**10.385.4.44 VTK\_LEGACY()** [1/4]

```
vtkImageColorViewer::VTK_LEGACY (
    int GetWholeZMin() )
```



**10.385.4.45** `VTK_LEGACY()` [2/4]

```
vtkImageColorViewer::VTK_LEGACY (
    int  GetWholeZMax() )
```

**10.385.4.46** `VTK_LEGACY()` [3/4]

```
vtkImageColorViewer::VTK_LEGACY (
    int  GetZSlice() )
```

**10.385.4.47** `VTK_LEGACY()` [4/4]

```
vtkImageColorViewer::VTK_LEGACY (
    void  SetZSliceint )
```

**10.385.4.48** `vtkBooleanMacro()`

```
vtkImageColorViewer::vtkBooleanMacro (
    OffScreenRendering ,
    int )
```

**10.385.4.49** `vtkGetMacro()` [1/2]

```
vtkImageColorViewer::vtkGetMacro (
    SliceOrientation ,
    int )
```

**10.385.4.50** `vtkGetMacro()` [2/2]

```
vtkImageColorViewer::vtkGetMacro (
    Slice ,
    int )
```

**10.385.4.51 vtkGetObjectMacro()** [1/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    RenderWindow ,
    vtkRenderWindow )
```

**10.385.4.52 vtkGetObjectMacro()** [2/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    Renderer ,
    vtkRenderer )
```

**10.385.4.53 vtkGetObjectMacro()** [3/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    ImageActor ,
    vtkImageActor )
```

**10.385.4.54 vtkGetObjectMacro()** [4/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    WindowLevel ,
    vtkImageMapToWindowLevelColors2 )
```

**10.385.4.55 vtkGetObjectMacro()** [5/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
    InteractorStyle ,
    vtkInteractorStyleImage )
```

**10.385.4.56 vtkTypeMacro()**

```
vtkImageColorViewer::vtkTypeMacro (
    vtkImageColorViewer ,
    vtkObject )
```

## 10.385.5 Friends And Related Function Documentation

### 10.385.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback [friend]
```

## 10.385.6 Member Data Documentation

### 10.385.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender [protected]
```

### 10.385.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor [protected]
```

### 10.385.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]
```

### 10.385.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]
```

### 10.385.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]
```

#### 10.385.6.6 **Renderer**

```
vtkRenderer* vtkImageColorViewer::Renderer [protected]
```

#### 10.385.6.7 **RenderWindow**

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]
```

#### 10.385.6.8 **Slice**

```
int vtkImageColorViewer::Slice [protected]
```

#### 10.385.6.9 **SliceOrientation**

```
int vtkImageColorViewer::SliceOrientation [protected]
```

#### 10.385.6.10 **WindowLevel**

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]
```

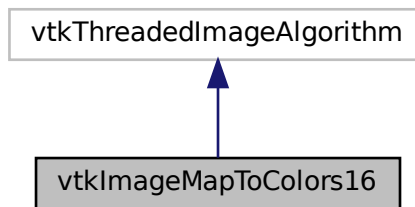
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

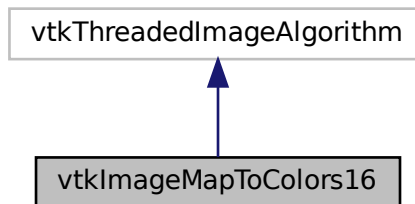
## 10.386 vtkImageMapToColors16 Class Reference

```
#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](#) ([OutputFormat](#), int)
- [vtkGetMacro](#) ([ActiveComponent](#), int)

- [vtkGetMacro](#) ([PassAlphaToOutput](#), int)
- [vtkGetObjectMacro](#) ([LookupTable](#), [vtkScalarsToColors](#))
- [vtkSetMacro](#) ([OutputFormat](#), int)
- [vtkSetMacro](#) ([ActiveComponent](#), 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](#)

## 10.386.1 Constructor & Destructor Documentation

### 10.386.1.1 [vtkImageMapToColors16\(\)](#)

```
vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]
```

### 10.386.1.2 [~vtkImageMapToColors16\(\)](#)

```
vtkImageMapToColors16::~~vtkImageMapToColors16 ( ) [protected]
```

## 10.386.2 Member Function Documentation

### 10.386.2.1 GetMTime()

```
virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]
```

### 10.386.2.2 New()

```
static vtkImageMapToColors16* vtkImageMapToColors16::New ( ) [static]
```

### 10.386.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.386.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

### 10.386.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

**10.386.2.6 SetLookupTable()**

```
virtual void vtkImageMapToColors16::SetLookupTable (
    vtkScalarsToColors * ) [virtual]
```

**10.386.2.7 SetOutputFormatToLuminance()**

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]
```

**10.386.2.8 SetOutputFormatToLuminanceAlpha()**

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]
```

**10.386.2.9 SetOutputFormatToRGB()**

```
void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]
```

**10.386.2.10 SetOutputFormatToRGBA()**

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]
```

**10.386.2.11 ThreadedRequestData()**

```
void vtkImageMapToColors16::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```



**10.386.2.12 vtkBooleanMacro()**

```
vtkImageMapToColors16::vtkBooleanMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.13 vtkGetMacro()** [1/3]

```
vtkImageMapToColors16::vtkGetMacro (
    OutputFormat ,
    int )
```

**10.386.2.14 vtkGetMacro()** [2/3]

```
vtkImageMapToColors16::vtkGetMacro (
    ActiveComponent ,
    int )
```

**10.386.2.15 vtkGetMacro()** [3/3]

```
vtkImageMapToColors16::vtkGetMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.16 vtkGetObjectMacro()**

```
vtkImageMapToColors16::vtkGetObjectMacro (
    LookupTable ,
    vtkScalarsToColors )
```

**10.386.2.17 vtkSetMacro()** [1/3]

```
vtkImageMapToColors16::vtkSetMacro (
    OutputFormat ,
    int )
```

**10.386.2.18 vtkSetMacro()** [2/3]

```
vtkImageMapToColors16::vtkSetMacro (
    ActiveComponent ,
    int )
```

**10.386.2.19 vtkSetMacro()** [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
    PassAlphaToOutput ,
    int )
```

**10.386.2.20 vtkTypeMacro()**

```
vtkImageMapToColors16::vtkTypeMacro (
    vtkImageMapToColors16 ,
    vtkThreadedImageAlgorithm )
```

**10.386.3 Member Data Documentation****10.386.3.1 ActiveComponent**

```
int vtkImageMapToColors16::ActiveComponent [protected]
```

**10.386.3.2 DataWasPassed**

```
int vtkImageMapToColors16::DataWasPassed [protected]
```

**10.386.3.3 LookupTable**

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
```

#### 10.386.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat [protected]
```

#### 10.386.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput [protected]
```

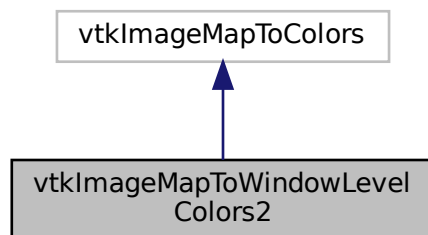
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

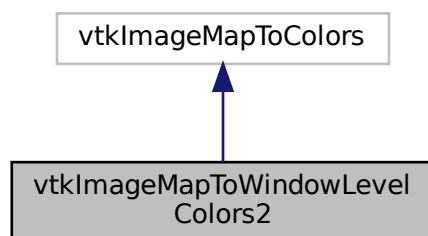
### 10.387 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



## Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkGetMacro](#) ([Window](#), double)
- [vtkGetMacro](#) ([Level](#), double)
- [vtkSetMacro](#) ([Window](#), double)
- [vtkSetMacro](#) ([Level](#), 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](#)

### 10.387.1 Constructor & Destructor Documentation

#### 10.387.1.1 [vtkImageMapToWindowLevelColors2](#)()

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]
```

#### 10.387.1.2 [~vtkImageMapToWindowLevelColors2](#)()

```
vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 ( ) [protected]
```

## 10.387.2 Member Function Documentation

### 10.387.2.1 New()

```
static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( ) [static]
```

### 10.387.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.387.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector ) [protected], [virtual]
```

### 10.387.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

### 10.387.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector ** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData *** inData,
    vtkImageData ** outData,
    int extent[6],
    int id ) [protected]
```

**10.387.2.6 vtkGetMacro()** [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Window ,
    double )
```

**10.387.2.7 vtkGetMacro()** [2/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
    Level ,
    double )
```

**10.387.2.8 vtkSetMacro()** [1/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Window ,
    double )
```

**10.387.2.9 vtkSetMacro()** [2/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
    Level ,
    double )
```

**10.387.2.10 vtkTypeMacro()**

```
vtkImageMapToWindowLevelColors2::vtkTypeMacro (
    vtkImageMapToWindowLevelColors2 ,
    vtkImageMapToColors )
```

**10.387.3 Member Data Documentation**

### 10.387.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level [protected]
```

### 10.387.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window [protected]
```

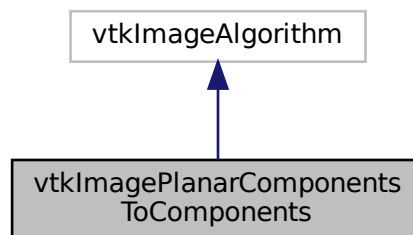
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

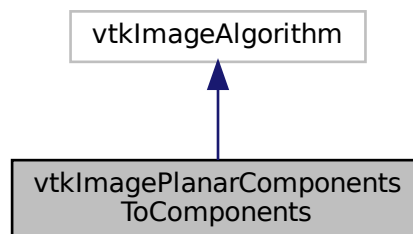
## 10.388 vtkImagePlanarComponentsToComponents Class Reference

```
#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 \*)

## 10.388.1 Constructor & Destructor Documentation

### 10.388.1.1 [vtkImagePlanarComponentsToComponents\(\)](#)

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( ) [protected]
```

### 10.388.1.2 [~vtkImagePlanarComponentsToComponents\(\)](#)

```
vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ( ) [inline], [protected]
```

## 10.388.2 Member Function Documentation

### 10.388.2.1 [New\(\)](#)

```
static vtkImagePlanarComponentsToComponents\* vtkImagePlanarComponentsToComponents::New ( ) [static]
```



## 10.388.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

## 10.388.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
    vtkInformation * ,
    vtkInformationVector ** ,
    vtkInformationVector * ) [protected], [virtual]
```

## 10.388.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
    vtkImagePlanarComponentsToComponents ,
    vtkImageAlgorithm )
```

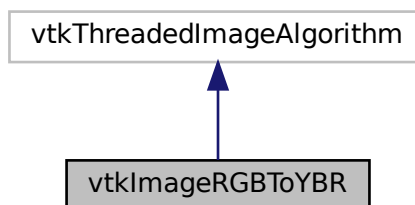
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

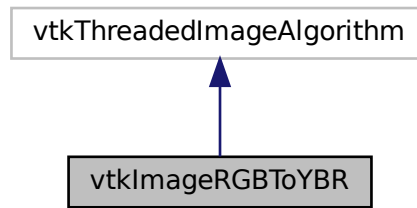
## 10.389 vtkImageRGBToYBR Class Reference

```
#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)

## 10.389.1 Constructor & Destructor Documentation

### 10.389.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]
```

### 10.389.1.2 ~vtkImageRGBToYBR()

```
vtkImageRGBToYBR::~~vtkImageRGBToYBR ( ) [inline], [protected]
```

## 10.389.2 Member Function Documentation

### 10.389.2.1 New()

```
static vtkImageRGBToYBR\* vtkImageRGBToYBR::New ( ) [static]
```

### 10.389.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.389.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.389.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
    vtkImageRGBToYBR ,
    vtkThreadedImageAlgorithm )
```

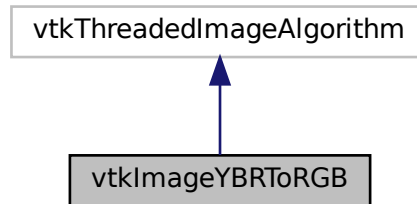
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

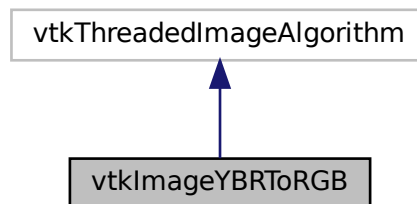
## 10.390 vtkImageYBRToRGB Class Reference

```
#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)

## 10.390.1 Constructor & Destructor Documentation

### 10.390.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]
```

### 10.390.1.2 ~vtkImageYBRToRGB()

```
vtkImageYBRToRGB::~~vtkImageYBRToRGB ( ) [inline], [protected]
```

## 10.390.2 Member Function Documentation

### 10.390.2.1 New()

```
static vtkImageYBRToRGB\* vtkImageYBRToRGB::New ( ) [static]
```

### 10.390.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

### 10.390.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
    vtkImageData * inData,
    vtkImageData * outData,
    int ext[6],
    int id ) [protected]
```

### 10.390.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
    vtkImageYBRToRGB ,
    vtkThreadedImageAlgorithm )
```

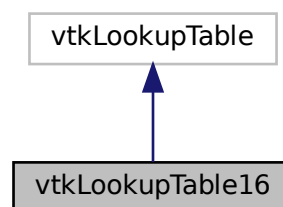
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

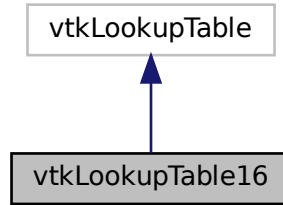
## 10.391 vtkLookupTable16 Class Reference

```
#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 numberOfValues, int inputIncrement, int outputFormat)

### Protected Attributes

- vtkUnsignedShortArray \* [Table16](#)

#### 10.391.1 Constructor & Destructor Documentation

#### 10.391.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
    int size = 256,
    int ext = 256 ) [protected]
```

#### 10.391.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~~vtkLookupTable16 ( ) [protected]
```

### 10.391.2 Member Function Documentation

#### 10.391.2.1 Build()

```
void vtkLookupTable16::Build ( )
```

#### 10.391.2.2 GetPointer()

```
unsigned short* vtkLookupTable16::GetPointer (
    const vtkIdType id ) [inline]
```

#### 10.391.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
    void * input,
    unsigned char * output,
    int inputDataType,
    int numberOfValues,
    int inputIncrement,
    int outputFormat ) [protected]
```



#### 10.391.2.4 New()

```
static vtkLookupTable16* vtkLookupTable16::New ( ) [static]
```

#### 10.391.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
    ostream & os,
    vtkIndent indent )
```

#### 10.391.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
    vtkIdType number )
```

#### 10.391.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (
    vtkLookupTable16 ,
    vtkLookupTable )
```

#### 10.391.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
    const vtkIdType id,
    const int number ) [inline]
```

### 10.391.3 Member Data Documentation

#### 10.391.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]
```

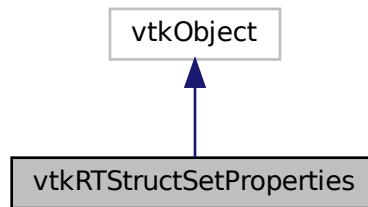
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

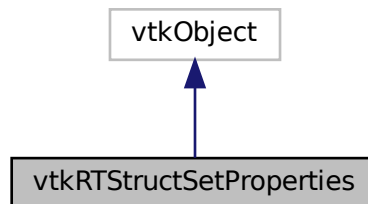
## 10.392 vtkRTStructSetProperties Class Reference

```
#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 \*reframerefid, const char \*roiname, const char \*ROI↔  
GenerationAlgorithm, 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](#) (StructureSetLabel)
- [vtkGetStringMacro](#) (StructureSetName)
- [vtkGetStringMacro](#) (StructureSetDate)
- [vtkGetStringMacro](#) (StructureSetTime)
- [vtkGetStringMacro](#) (SOPInstanceUID)
- [vtkGetStringMacro](#) (StudyInstanceUID)
- [vtkGetStringMacro](#) (SeriesInstanceUID)
- [vtkGetStringMacro](#) (ReferenceSeriesInstanceUID)
- [vtkGetStringMacro](#) (ReferenceFrameOfReferenceUID)
- [vtkSetStringMacro](#) (StructureSetLabel)
- [vtkSetStringMacro](#) (StructureSetName)
- [vtkSetStringMacro](#) (StructureSetDate)
- [vtkSetStringMacro](#) (StructureSetTime)
- [vtkSetStringMacro](#) (SOPInstanceUID)
- [vtkSetStringMacro](#) (StudyInstanceUID)
- [vtkSetStringMacro](#) (SeriesInstanceUID)
- [vtkSetStringMacro](#) (ReferenceSeriesInstanceUID)
- [vtkSetStringMacro](#) (ReferenceFrameOfReferenceUID)
- [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](#)

### 10.392.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#).

### 10.392.2 Constructor & Destructor Documentation

#### 10.392.2.1 vtkRTStructSetProperties()

```
vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]
```

#### 10.392.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~~vtkRTStructSetProperties ( ) [protected]
```

### 10.392.3 Member Function Documentation

#### 10.392.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
    vtkIdType pdnum,
    const char * classuid,
    const char * instanceuid )
```

### 10.392.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
    const char * classuid,
    const char * instanceuid )
```

### 10.392.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
    int roinumber,
    const char * refframerefuid,
    const char * roiname,
    const char * ROIGenerationAlgorithm,
    const char * ROIDescription = 0 )
```

### 10.392.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
    int refnumber,
    int observationnumber,
    const char * rtroiinterpretedtype,
    const char * roiinterpreter,
    const char * roiobservationlabel = 0 )
```

### 10.392.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ( ) [virtual]
```

### 10.392.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
    vtkRTStructSetProperties * p ) [virtual]
```

**10.392.3.7 GetContourReferencedFrameOfReferenceClassUID()**

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
    vtkIdType pdnum,
    vtkIdType id )
```

**10.392.3.8 GetContourReferencedFrameOfReferenceInstanceUID()**

```
const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
    vtkIdType pdnum,
    vtkIdType id )
```

**10.392.3.9 GetNumberOfContourReferencedFrameOfReferences()** [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
```

**10.392.3.10 GetNumberOfContourReferencedFrameOfReferences()** [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
    vtkIdType pdnum )
```

**10.392.3.11 GetNumberOfReferencedFrameOfReferences()**

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
```

**10.392.3.12 GetNumberOfStructureSetROIs()**

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
```

**10.392.3.13 GetReferencedFrameOfReferenceClassUID()**

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
    vtkIdType id )
```

**10.392.3.14 GetReferencedFrameOfReferenceInstanceUID()**

```
const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
    vtkIdType id )
```

**10.392.3.15 GetStructureSetObservationNumber()**

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
    vtkIdType id )
```

**10.392.3.16 GetStructureSetROIDescription()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIDescription (
    vtkIdType id )
```

**10.392.3.17 GetStructureSetROIGenerationAlgorithm()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
    vtkIdType )
```

**10.392.3.18 GetStructureSetROIName()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIName (
    vtkIdType )
```

**10.392.3.19 GetStructureSetROINumber()**

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
    vtkIdType id )
```

**10.392.3.20 GetStructureSetROIObservationLabel()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
    vtkIdType id )
```

**10.392.3.21 GetStructureSetROIRefFrameRefUID()**

```
const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
    vtkIdType )
```

**10.392.3.22 GetStructureSetRTROIInterpretedType()**

```
const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
    vtkIdType id )
```

**10.392.3.23 New()**

```
static vtkRTStructSetProperties\* vtkRTStructSetProperties::New ( ) [static]
```

**Examples:**

[GenerateRTSTRUCT.cxx](#).

**10.392.3.24 PrintSelf()**

```
void vtkRTStructSetProperties::PrintSelf (
    ostream & os,
    vtkIndent indent )
```



**10.392.3.25 vtkGetStringMacro()** [1/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetLabel )
```

**10.392.3.26 vtkGetStringMacro()** [2/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetName )
```

**10.392.3.27 vtkGetStringMacro()** [3/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetDate )
```

**10.392.3.28 vtkGetStringMacro()** [4/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StructureSetTime )
```

**10.392.3.29 vtkGetStringMacro()** [5/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SOPInstanceUID )
```

**10.392.3.30 vtkGetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    StudyInstanceUID )
```

**10.392.3.31 vtkGetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    SeriesInstanceUID )
```

**10.392.3.32 vtkGetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.392.3.33 vtkGetStringMacro()** [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
    ReferenceFrameOfReferenceUID )
```

**10.392.3.34 vtkSetStringMacro()** [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetLabel )
```

**10.392.3.35 vtkSetStringMacro()** [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetName )
```

**10.392.3.36 vtkSetStringMacro()** [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetDate )
```

**10.392.3.37 vtkSetStringMacro()** [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StructureSetTime )
```

**10.392.3.38 vtkSetStringMacro()** [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SOPInstanceUID )
```

**10.392.3.39 vtkSetStringMacro()** [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    StudyInstanceUID )
```

**10.392.3.40 vtkSetStringMacro()** [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    SeriesInstanceUID )
```

**10.392.3.41 vtkSetStringMacro()** [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceSeriesInstanceUID )
```

**10.392.3.42 vtkSetStringMacro()** [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
    ReferenceFrameOfReferenceUID )
```

### 10.392.3.43 vtkTypeMacro()

```
vtkRTStructSetProperties::vtkTypeMacro (
    vtkRTStructSetProperties ,
    vtkObject )
```

## 10.392.4 Member Data Documentation

### 10.392.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]
```

### 10.392.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]
```

### 10.392.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]
```

### 10.392.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID [protected]
```

### 10.392.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID [protected]
```

#### 10.392.4.6 StructureSetDate

char\* vtkRTStructSetProperties::StructureSetDate [protected]

#### 10.392.4.7 StructureSetLabel

char\* vtkRTStructSetProperties::StructureSetLabel [protected]

#### 10.392.4.8 StructureSetName

char\* vtkRTStructSetProperties::StructureSetName [protected]

#### 10.392.4.9 StructureSetTime

char\* vtkRTStructSetProperties::StructureSetTime [protected]

#### 10.392.4.10 StudyInstanceUID

char\* vtkRTStructSetProperties::StudyInstanceUID [protected]

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

- [vtkRTStructSetProperties.h](#)

## 10.393 gdcM::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcMWaveform.h>
```

### Public Member Functions

- [Waveform](#) ()

### 10.393.1 Detailed Description

[Waveform](#) class.

### 10.393.2 Constructor & Destructor Documentation

#### 10.393.2.1 Waveform()

```
gdcM::Waveform::Waveform ( ) [inline]
```

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

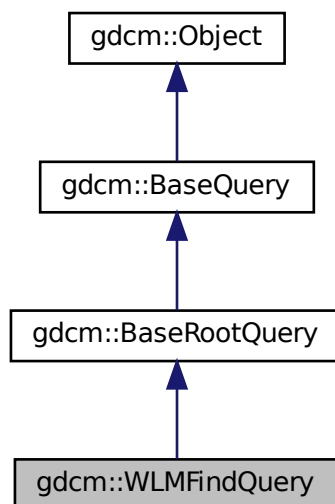
- [gdcMWaveform.h](#)

## 10.394 gdcM::WLMFindQuery Class Reference

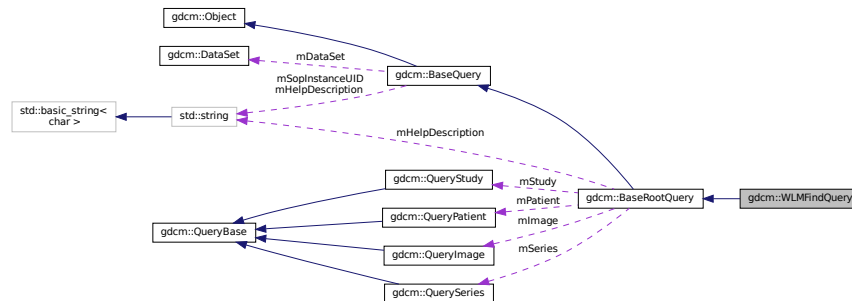
PatientRootQuery.

```
#include <gdcMWLMFindQuery.h>
```

Inheritance diagram for gdcM::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



## Public Member Functions

- [WLMFindQuery](#) ()
- [UIDs::TSName GetAbstractSyntaxUID](#) () const
- [std::vector< Tag > GetTagListByLevel](#) (const [EQueryLevel](#) &inQueryLevel)
- void [InitializeDataSet](#) (const [EQueryLevel](#) &inQueryLevel)
- bool [ValidateQuery](#) (bool inStrict=true) const

## Protected Member Functions

- [DataSet GetValidDataSet](#) () const

## Friends

- class [QueryFactory](#)

## Additional Inherited Members

### 10.394.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

### 10.394.2 Constructor & Destructor Documentation

### 10.394.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ( )
```

## 10.394.3 Member Function Documentation

### 10.394.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID ( ) const [virtual]
```

Implements [gdcm::BaseQuery](#).

### 10.394.3.2 GetTagListByLevel()

```
std::vector<Tag> gdcm::WLMFindQuery::GetTagListByLevel (
    const EQueryLevel & inQueryLevel ) [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](#).

### 10.394.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet ( ) const [protected]
```

### 10.394.3.4 InitializeDataSet()

```
void gdcm::WLMFindQuery::InitializeDataSet (
    const EQueryLevel & inQueryLevel ) [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](#).



## 10.394.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
    bool inStrict = true ) const [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](#).

## 10.394.4 Friends And Related Function Documentation

## 10.394.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

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

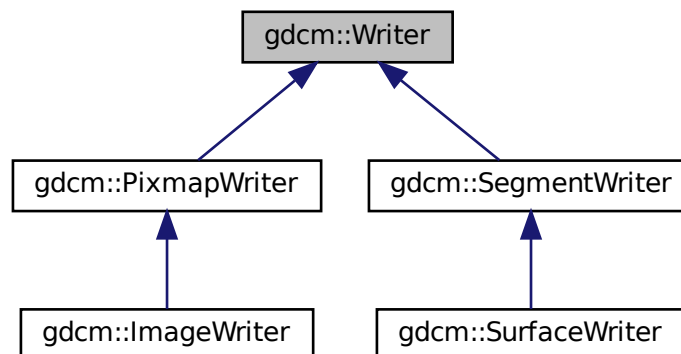
- [gdcmWLMFindQuery.h](#)

## 10.395 gdcm::Writer Class Reference

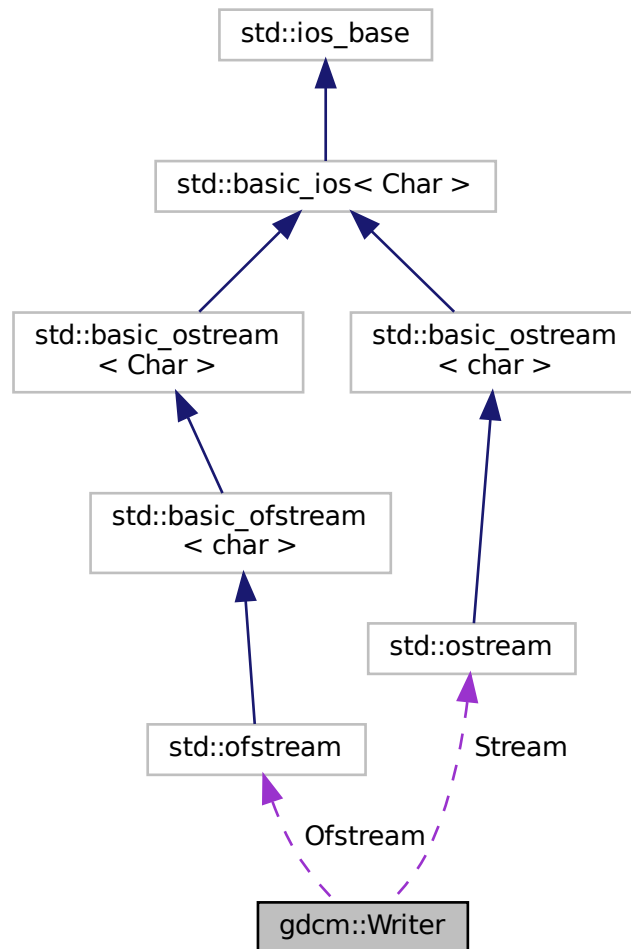
[Writer](#) ala DOM (Document [Object](#) Model)

```
#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](#)

## 10.395.1 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 (garanteed 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](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

## 10.395.2 Constructor & Destructor Documentation

### 10.395.2.1 Writer()

```
gdcm::Writer::Writer ( )
```

### 10.395.2.2 ~Writer()

```
virtual gdcm::Writer::~~Writer ( ) [virtual]
```

## 10.395.3 Member Function Documentation

### 10.395.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]
```

Examples:

[CreateFakeRTDOSE.cxx](#), [FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

### 10.395.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]
```

### 10.395.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation ( ) const [inline], [protected]
```

### 10.395.3.4 GetFile()

```
File& gdcm::Writer::GetFile ( ) [inline]
```

#### Examples:

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), [StreamImageReaderTest.cxx](#), and [TemplateEmptyImage.cxx](#).

### 10.395.3.5 GetStreamPtr()

```
std::ostream* gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]
```

### 10.395.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
    bool b ) [inline]
```

Undocumented function, do not use (= leave default)

#### Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

### 10.395.3.7 SetFile()

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

Set/Get the DICOM file ([DataSet](#) + Header)

#### Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

### 10.395.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
    const char * filename_native )
```

Set the filename of DICOM file to write:

#### Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [TemplateEmptyImage.cxx](#).

### 10.395.3.9 SetStream()

```
void gdcm::Writer::SetStream (
    std::ostream & output_stream ) [inline]
```

Set user ostream buffer.

### 10.395.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
    bool b ) [inline], [protected]
```

### 10.395.3.11 Write()

```
virtual bool gdcm::Writer::Write ( ) [virtual]
```

Main function to tell the writer to write.

Reimplemented in [gdcm::PixmapWriter](#), [gdcm::ImageWriter](#), [gdcm::SurfaceWriter](#), and [gdcm::SegmentWriter](#).

#### Examples:

[ChangePrivateTags.cxx](#), [ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateFakeRTDOSE.cxx](#), [CreateJPIPDataSet.cxx](#), [DeriveSeries.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAI Bug JPEGLS.cxx](#), [FixOrientation.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MakeTemplate.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), and [rle2img.cxx](#).

## 10.395.4 Friends And Related Function Documentation

### 10.395.4.1 StreamImageWriter

```
friend class StreamImageWriter [friend]
```

## 10.395.5 Member Data Documentation

### 10.395.5.1 Ofstream

```
std::ofstream* gdcm::Writer::Ofstream [protected]
```

### 10.395.5.2 Stream

```
std::ostream* gdcm::Writer::Stream [protected]
```

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

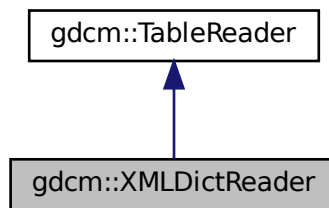
- [gdcmWriter.h](#)

## 10.396 gdcm::XMLDictReader Class Reference

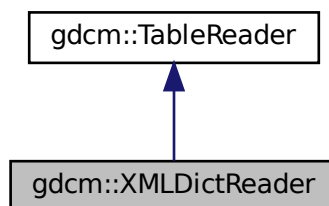
Class for representing a [XMLDictReader](#).

```
#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)



## Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

### 10.396.1 Detailed Description

Class for representing a [XMLDictReader](#).

#### Note

bla Will read the DICOMV3.xml file

### 10.396.2 Constructor & Destructor Documentation

#### 10.396.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ( )
```

#### 10.396.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~XMLDictReader ( ) [inline]
```

### 10.396.3 Member Function Documentation

#### 10.396.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.396.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.396.3.3 GetDict()

```
const Dict& gdcm::XMLDictReader::GetDict ( ) [inline]
```

### 10.396.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.396.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.396.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
    const char * name,
    const char ** atts ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

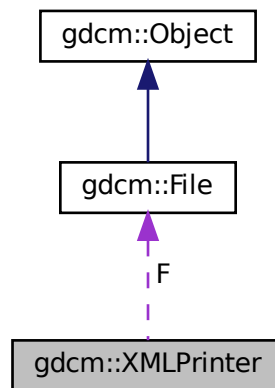
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

## 10.397 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](#)

## 10.397.1 Member Enumeration Documentation

### 10.397.1.1 PrintStyles

```
enum gdcm::XMLPrinter::PrintStyles
```

#### Enumerator

OnlyUUID	
LOADBULKDATA	

## 10.397.2 Constructor & Destructor Documentation

### 10.397.2.1 XMLPrinter()

```
gdcm::XMLPrinter::XMLPrinter ( )
```

### 10.397.2.2 ~XMLPrinter()

```
virtual gdcm::XMLPrinter::~~XMLPrinter ( ) [virtual]
```

## 10.397.3 Member Function Documentation

### 10.397.3.1 GetPrintStyle()

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]
```

### 10.397.3.2 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

### 10.397.3.3 Print()

```
void gdcm::XMLPrinter::Print (
    std::ostream & os )
```

### 10.397.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (
    std::ostream & os,
    const Dicts & dicts,
    const DataSet & ds,
    const DataElement & de,
    const TransferSyntax & ts ) [protected]
```

### 10.397.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (
    const DataSet & ds,
    const TransferSyntax & ts,
    std::ostream & os )
```

### 10.397.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (
    const SequenceOfItems * sqi,
    const TransferSyntax & ts,
    std::ostream & os ) [protected]
```

### 10.397.3.7 SetFile()

```
void gdcM::XMLPrinter::SetFile (
    File const & f ) [inline]
```

### 10.397.3.8 SetStyle()

```
void gdcM::XMLPrinter::SetStyle (
    PrintStyles ps ) [inline]
```

## 10.397.4 Member Data Documentation

### 10.397.4.1 F

```
const File* gdcM::XMLPrinter::F [protected]
```

### 10.397.4.2 PrintStyle

```
PrintStyles gdcM::XMLPrinter::PrintStyle [protected]
```

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

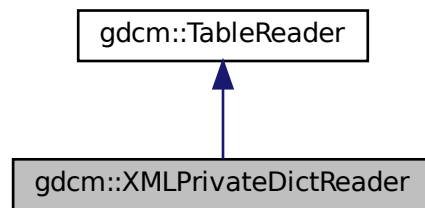
- [gdcMXMLPrinter.h](#)

## 10.398 gdcm::XMLPrivateDictReader Class Reference

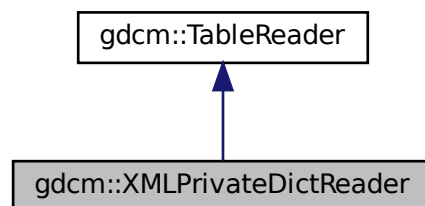
Class for representing a [XMLPrivateDictReader](#).

```
#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)

## Protected Member Functions

- void [HandleDescription](#) (const char \*\*atts)
- void [HandleEntry](#) (const char \*\*atts)

### 10.398.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

#### Note

bla Will read the Private.xml file

### 10.398.2 Constructor & Destructor Documentation

#### 10.398.2.1 XMLPrivateDictReader()

```
gdcM::XMLPrivateDictReader::XMLPrivateDictReader ( )
```

#### 10.398.2.2 ~XMLPrivateDictReader()

```
gdcM::XMLPrivateDictReader::~~XMLPrivateDictReader ( ) [inline]
```

### 10.398.3 Member Function Documentation

#### 10.398.3.1 CharacterDataHandler()

```
void gdcM::XMLPrivateDictReader::CharacterDataHandler (
    const char * data,
    int length ) [virtual]
```

Reimplemented from [gdcM::TableReader](#).



### 10.398.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
    const char * name ) [virtual]
```

Reimplemented from [gdcm::TableReader](#).

### 10.398.3.3 GetPrivateDict()

```
const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]
```

### 10.398.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
    const char ** atts ) [protected]
```

### 10.398.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
    const char ** atts ) [protected]
```

### 10.398.3.6 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 11

# File Documentation

### 11.1 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"  
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



### Classes

- class `gdcm::network::AAabortPDU`  
*AAabortPDU*.

### Namespaces

- `gdcm`
- `gdcm::network`

## 11.2 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

- [gdcm](#)
- [gdcm::network](#)

## 11.3 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



## Classes

- class [gdcm::network::AAssociateRJPDU](#)  
*AAssociateRJPDU.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.4 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::AAssociateRQ](#)PDU  
*AAssociateRQ*PDU.

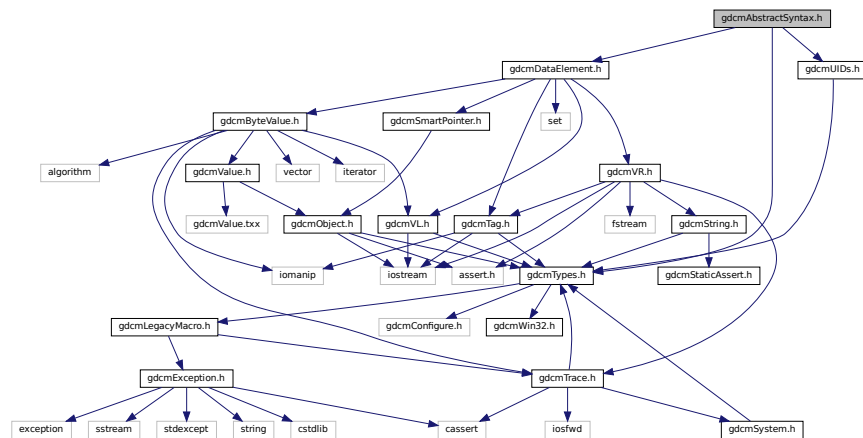
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

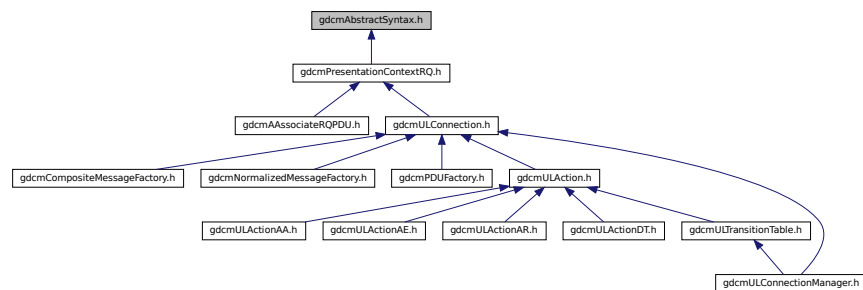
## 11.5 gdcmAbstractSyntax.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmAbstractSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::AbstractSyntax](#)  
*AbstractSyntax.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.6 gdcmAnonymizeEvent.h File Reference

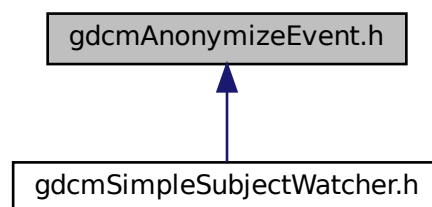
```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for gdcmAnonymizeEvent.h:



This graph shows which files directly or indirectly include this file:



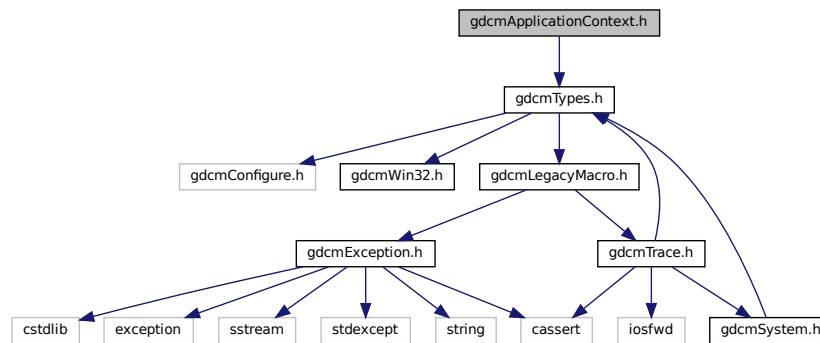




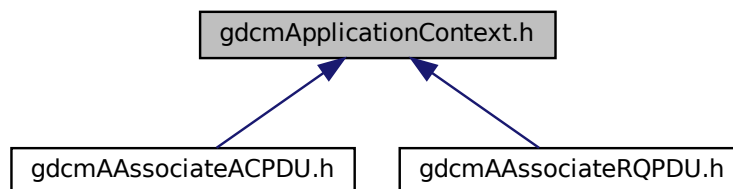
## 11.8 gdcmApplicationContext.h File Reference

```
#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

- `gdcm`
- `gdcm::network`

## 11.9 gdcApplicationEntity.h File Reference

```
#include "gdcTypes.h"
```

```
#include <vector>
```

```
#include <stdlib.h>
```

Include dependency graph for gdcApplicationEntity.h:



### Classes

- class `gdc::ApplicationEntity`  
*ApplicationEntity.*

### Namespaces

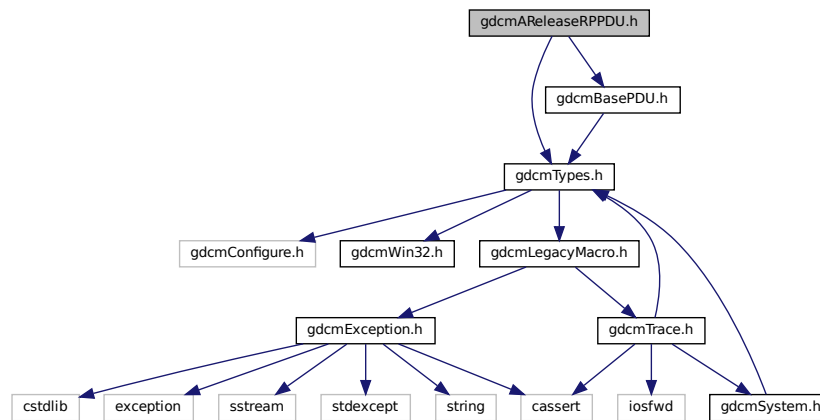
- `gdc`

## 11.10 gdcAReleaseRPPDU.h File Reference

```
#include "gdcTypes.h"
```

```
#include "gdcBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



## Classes

- class `gdcm::network::AReleaseRPPDU`  
*AReleaseRPPDU.*

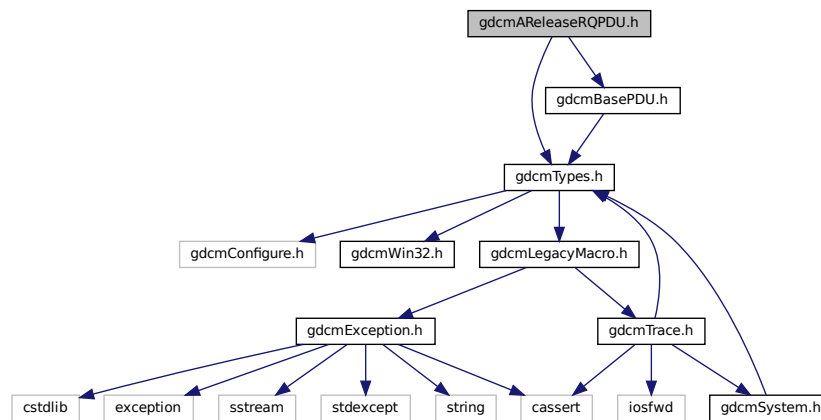
## Namespaces

- `gdcm`
- `gdcm::network`

## 11.11 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for `gdcmAReleaseRQPDU.h`:



## Classes

- class [gdcm::network::AReleaseRQPDU](#)  
*AReleaseRQPDU.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.12 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ARTIMTimer](#)  
[ARTIMTimer](#).

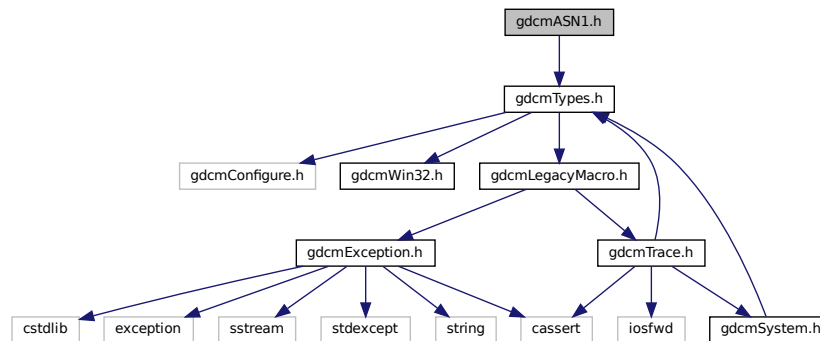
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.13 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



## Classes

- class [gdcm::ASN1](#)  
*Class for ASN1.*

## Namespaces

- [gdcm](#)

## 11.14 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



### Classes

- class [gdcm::network::AsynchronousOperationsWindowSub](#)  
*AsynchronousOperationsWindowSub.*

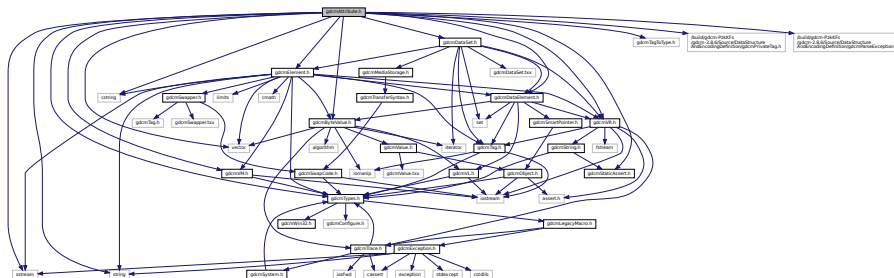
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.15 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"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
```

Include dependency graph for `gdcmAttribute.h`:



```
graph BT; gdcmspacing[gdcmspacing.h] --> gdcmattribute[gdcmAttribute.h];
```

- class `gdcmm::Attribute< Group, Element, TVR, TVM >`  
*Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary.*
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >`
- class `gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >`
- class `gdcmm::VRVLSize< T >`
- class `gdcmm::VRVLSize< 0 >`
- class `gdcmm::VRVLSize< 1 >`

- **gdcm**

## 11.16 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



### Classes

- class [gdcm::AudioCodec](#)  
*AudioCodec.*

### Namespaces

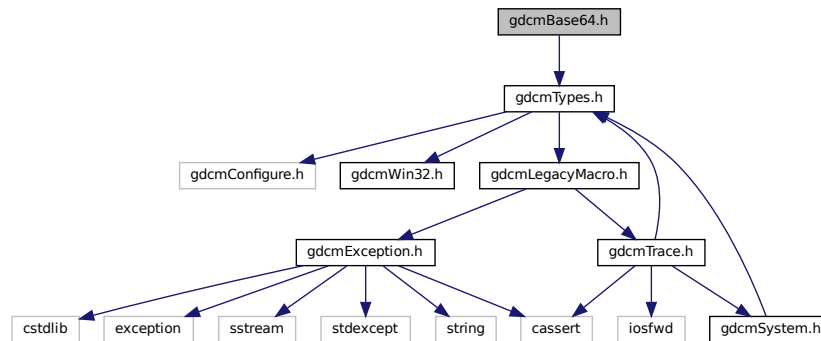
- [gdcm](#)

## 11.17 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```



Include dependency graph for gdcmbase64.h:



## Classes

- class `gdcm::Base64`  
*Class for Base64.*

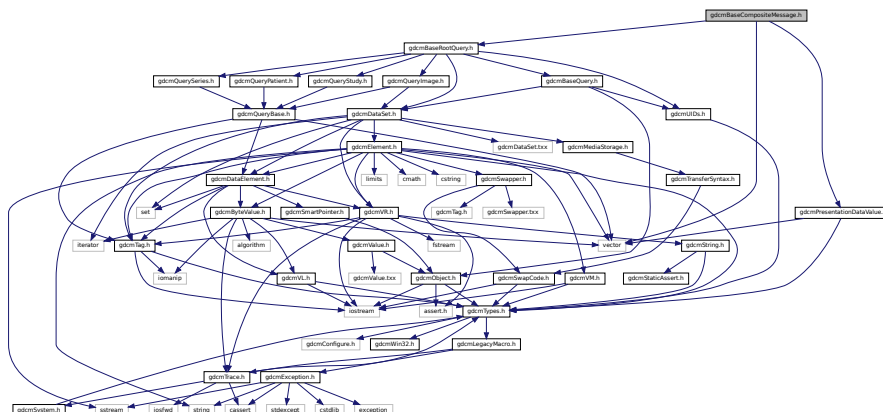
## Namespaces

- **gdcm**

## 11.18 gdcmBaseCompositeMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
```

Include dependency graph for `gdcmBaseCompositeMessage.h`:











## 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` }

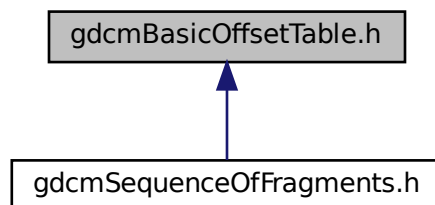
## 11.23 gdcmBasicOffsetTable.h File Reference

```
#include "gdcmFragment.h"
```

Include dependency graph for `gdcmBasicOffsetTable.h`:



This graph shows which files directly or indirectly include this file:









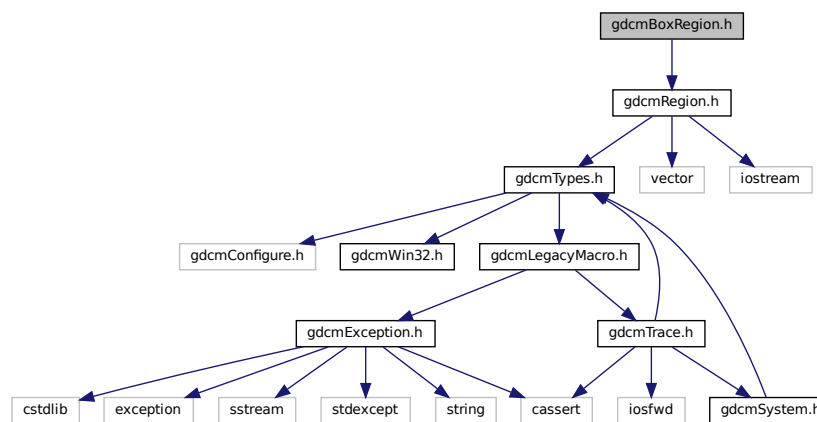
## Namespaces

- [gdcm](#)

## 11.26 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



## Classes

- class [gdcm::BoxRegion](#)  
Class for manipulation box region.

## Namespaces

- [gdcm](#)

## 11.27 gdcmByteBuffer.h File Reference

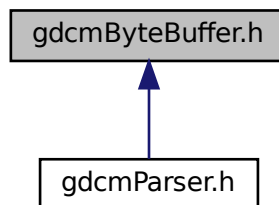
```
#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
```

```
#include <string.h>
```

Include dependency graph for `gdcmByteBuffer.h`:



This graph shows which files directly or indirectly include this file:



## Classes

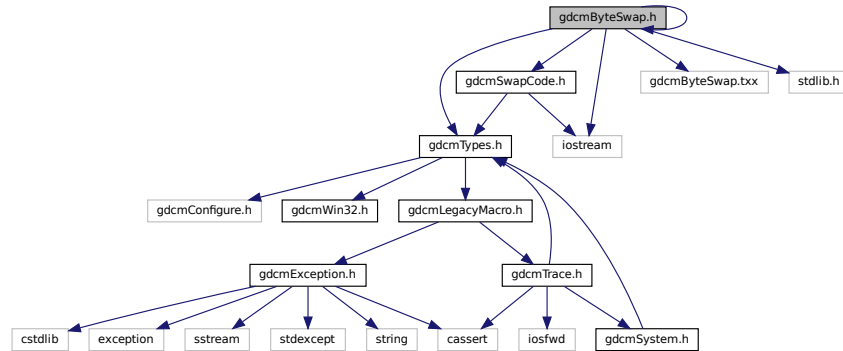
- class `gdcm::ByteBuffer`  
*ByteBuffer.*

## Namespaces

- `gdcm`

## 11.28 gdcmByteSwap.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.txx"
Include dependency graph for gdcmByteSwap.h:
```



### Classes

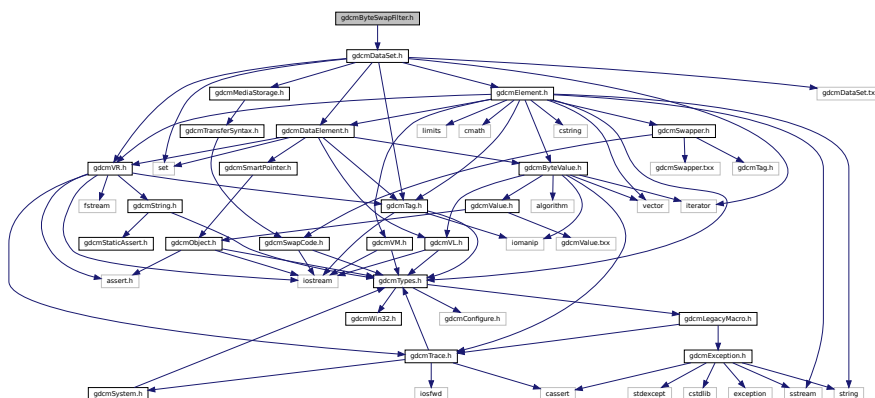
- class [gdcm::ByteSwap< T >](#)  
*ByteSwap.*

### Namespaces

- [gdcm](#)

## 11.29 gdcmByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
Include dependency graph for gdcmByteSwapFilter.h:
```



## Classes

- class [gdcm::ByteSwapFilter](#)  
*ByteSwapFilter.*

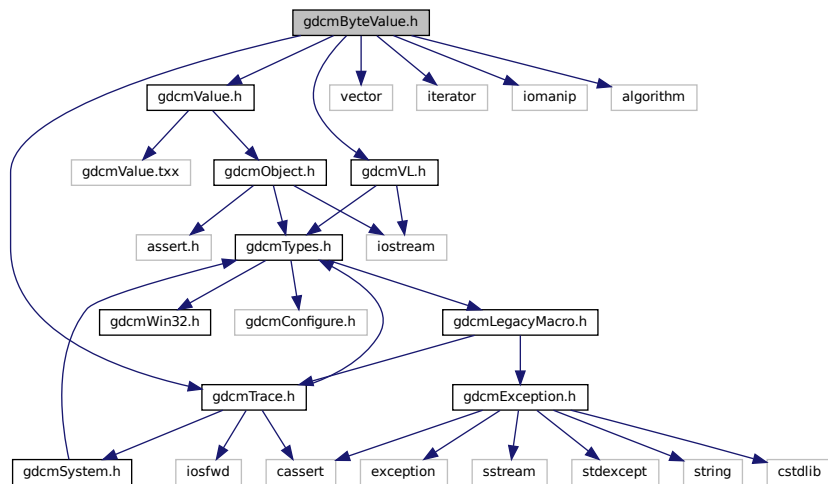
## Namespaces

- [gdcm](#)

## 11.30 gdcmByteValue.h File Reference

```
#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
#include <algorithm>
```

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)

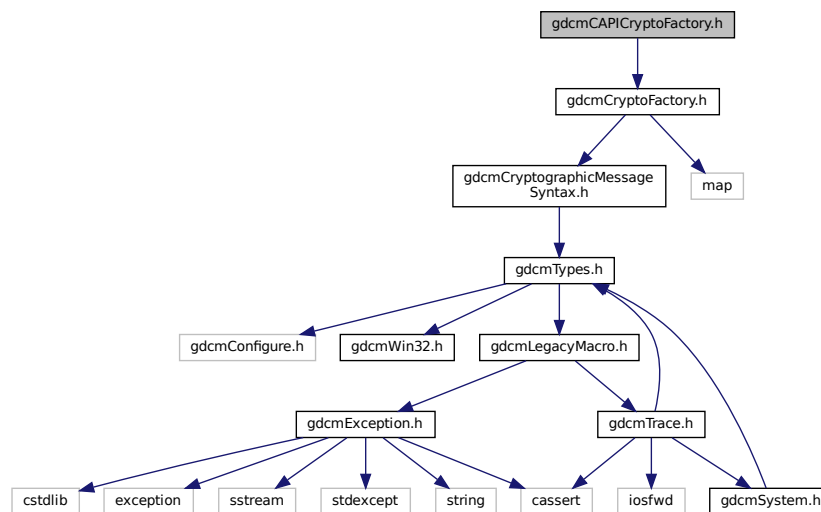
## Namespaces

- [gdcm](#)

## 11.31 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



## Classes

- class [gdcm::CAPICryptoFactory](#)

## Namespaces

- [gdcm](#)

## 11.32 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](#)

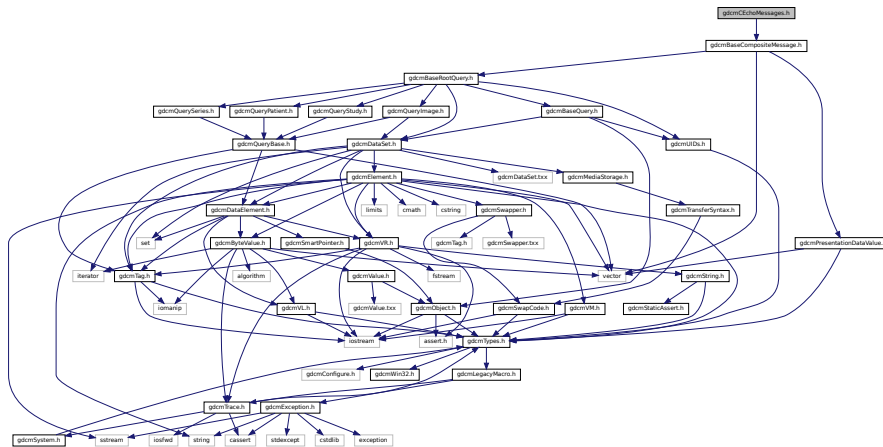
### Namespaces

- [gdcm](#)

## 11.33 gdcmCEchoMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

Include dependency graph for gdcmCEchoMessages.h:



## Classes

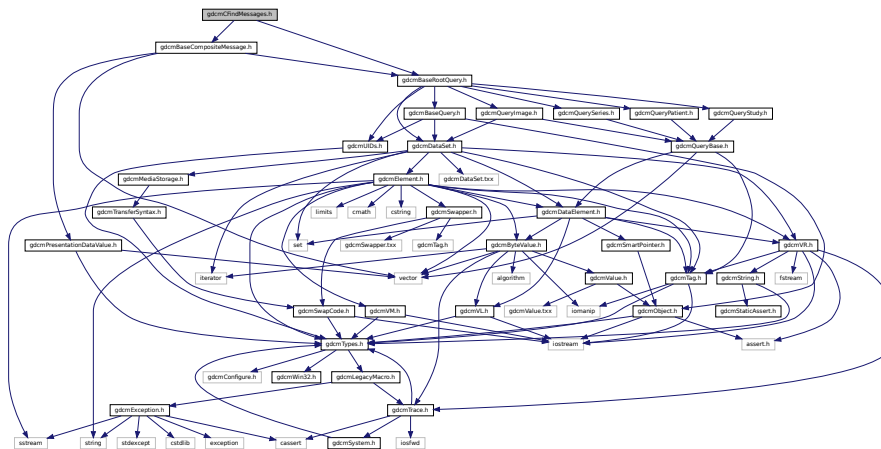
- class [gdcm::network::CEchoRQ](#)  
*CEchoRQ.*
- class [gdcm::network::CEchoRSP](#)  
*CEchoRSP* this file defines the messages for the cecho action.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.34 gdcmCFindMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```



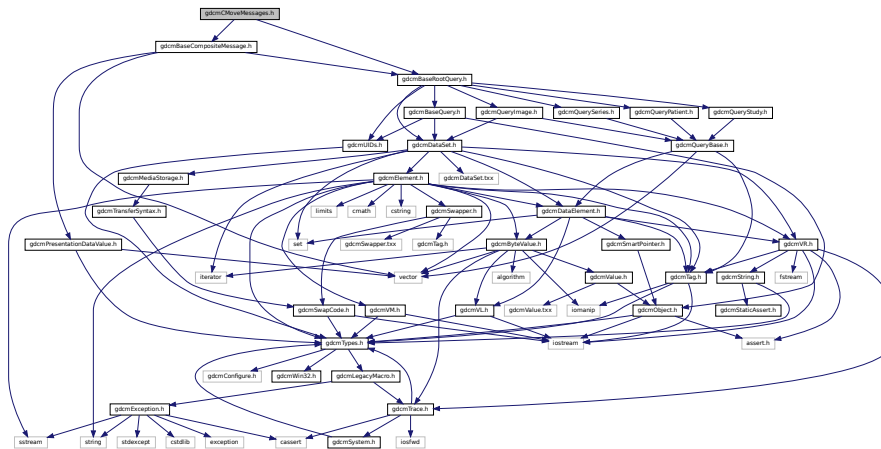
- class `gdcm::network::CFindCancelRQ`  
*CFindCancelRQ* this file defines the messages for the *cfind* action.
- class `gdcm::network::CFindRQ`  
*CFindRQ*.
- class `gdcm::network::CFindRSP`  
*CFindRSP* this file defines the messages for the *cfind* action.

- `gdcm`
- `gdcm::network`

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```



Include dependency graph for gdcmCMoveMessages.h:



## Classes

- class [gdcm::network::CMoveCancelRq](#)
- class [gdcm::network::CMoveRQ](#)  
[CMoveRQ](#).
- class [gdcm::network::CMoveRSP](#)  
[CMoveRSP](#) this file defines the messages for the cmove action.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

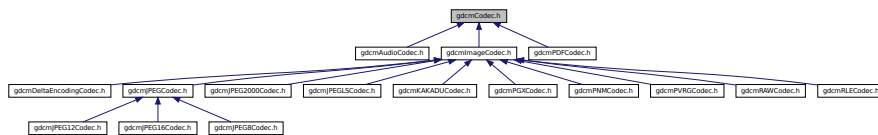
## 11.36 gdcmCodec.h File Reference

```
#include "gdcmCoder.h"
#include "gdcmDecoder.h"
```

Include dependency graph for `gdcmCodec.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Codec`  
*Codec class.*

## Namespaces

- `gdcm`

## 11.37 gdcmCoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCoder.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Coder](#)  
*Coder.*

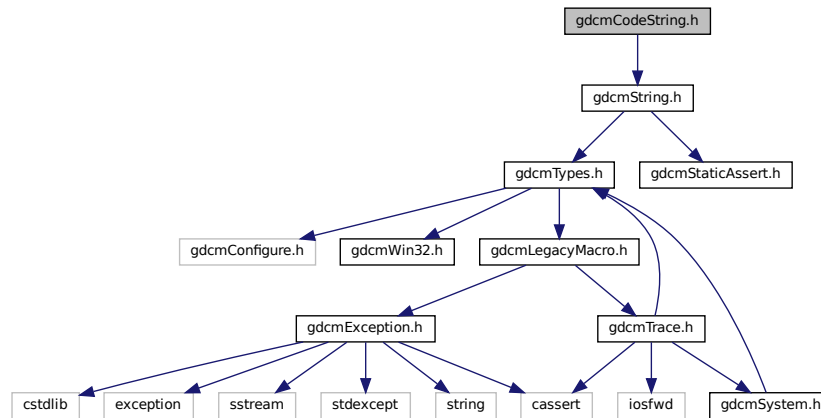
## Namespaces

- [gdcm](#)

## 11.38 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



### Classes

- class `gdcm::CodeString`  
*CodeString.*

### Namespaces

- `gdcm`

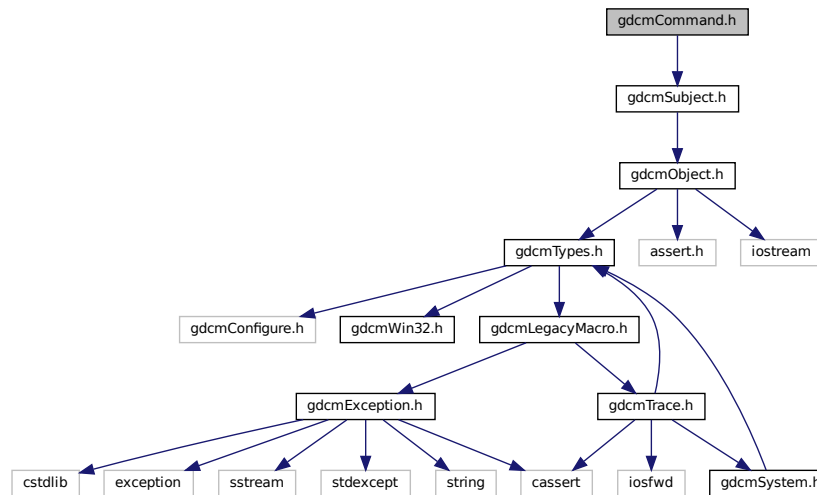
### Functions

- 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)

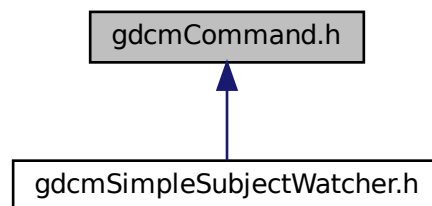
## 11.39 gdcmCommand.h File Reference

```
#include "gdcmSubject.h"
```

Include dependency graph for gdcmCommand.h:



This graph shows which files directly or indirectly include this file:

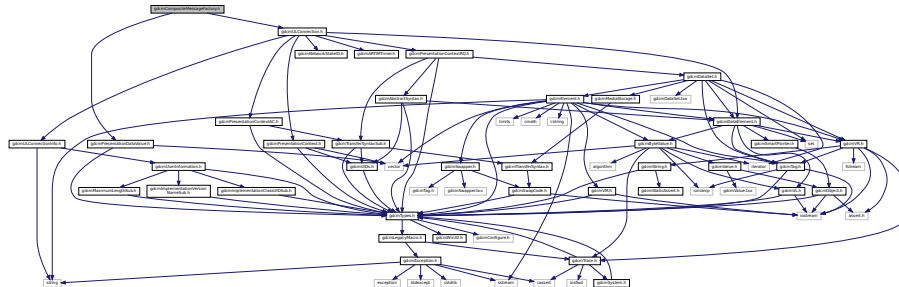


### Classes

- class `gdcm::Command`  
*Command superclass for callback/observer methods.*
- class `gdcm::MemberCommand< T >`  
*Command subclass that calls a pointer to a member function.*
- class `gdcm::SimpleMemberCommand< T >`  
*Command subclass that calls a pointer to a member function.*



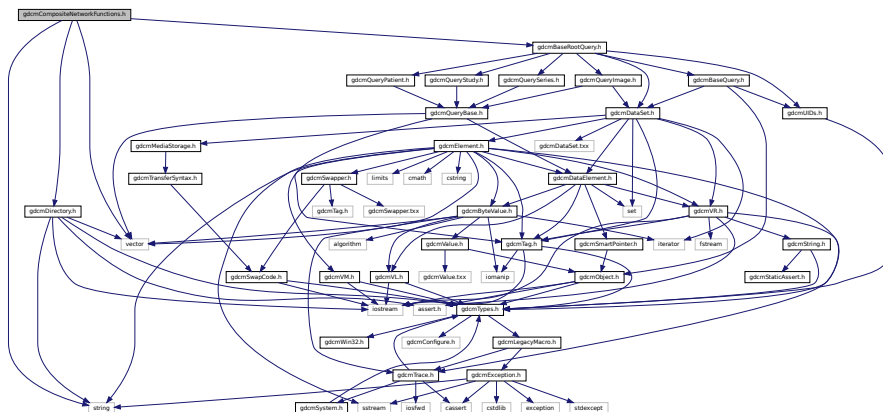
```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
Include dependency graph for gdcmCompositeMessageFactory.h:
```



- class `gdc::network::CompositeMessageFactory`  
*CompositeMessageFactory*.

- `gdcm`
- `gdcm::network`

```
#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
Include dependency graph for gdcmCompositeNetworkFunctions.h:
```



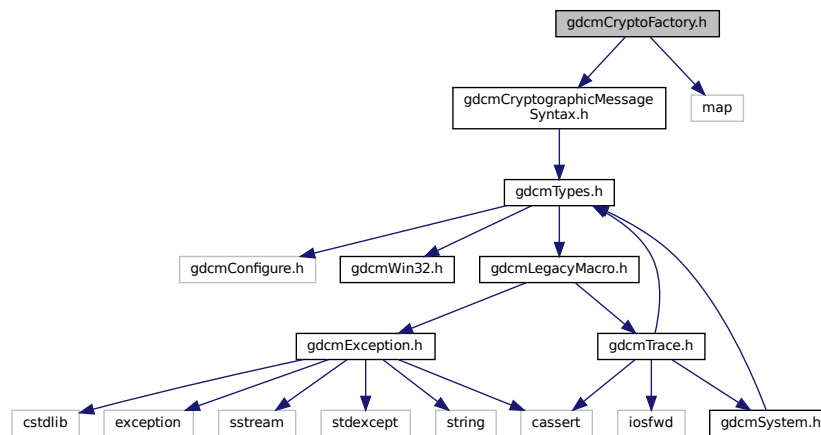




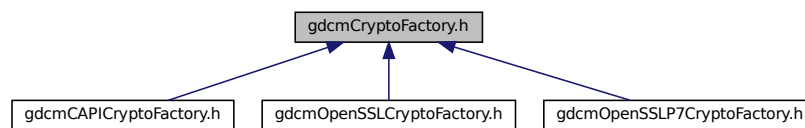
## 11.45 gdcmCryptoFactory.h File Reference

```
#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.

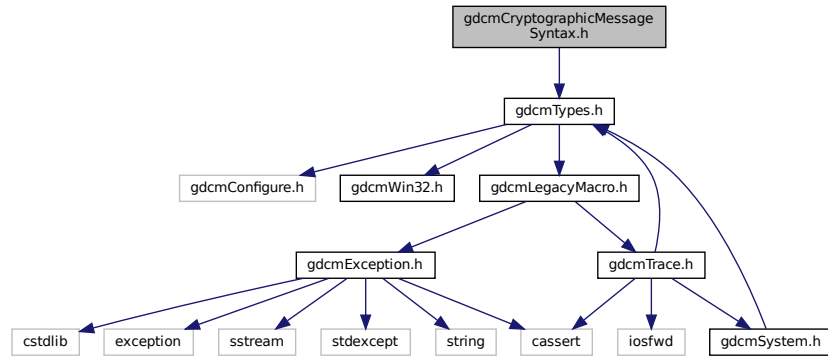
### Namespaces

- [gdcm](#)

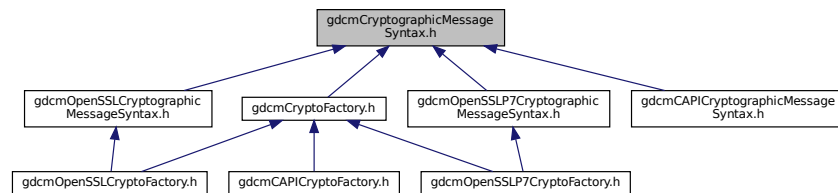
## 11.46 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



### Classes

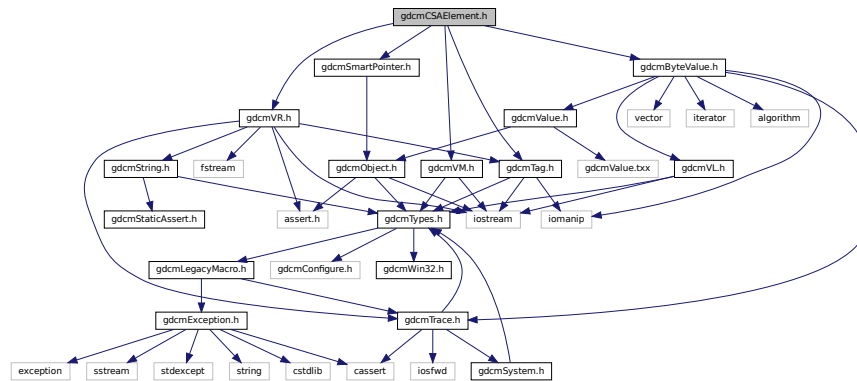
- class `gdcm::CryptographicMessageSyntax`

### Namespaces

- `gdcm`

## 11.47 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

- [gdcm](#)



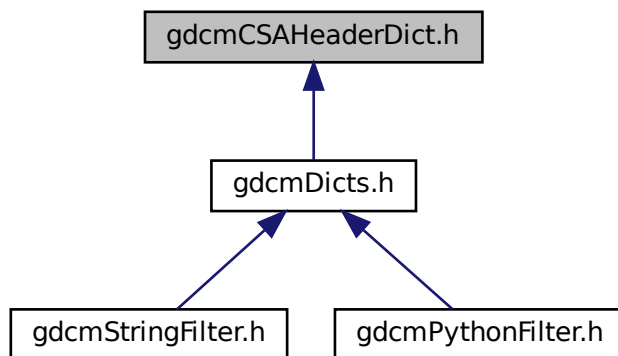
## 11.49 gdcmCSAHeaderDict.h File Reference

```
#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

- [gdcm](#)

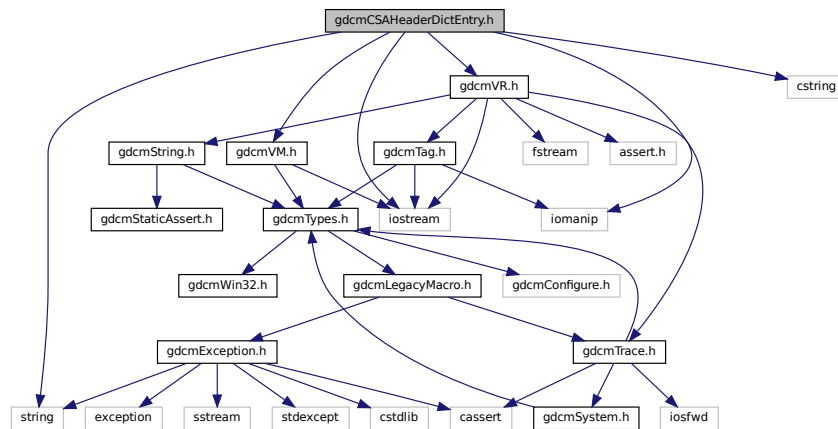
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

## 11.50 gdcmCSAHeaderDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <cstring>
```

Include dependency graph for `gdcmCSAHeaderDictEntry.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

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

## 11.51 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```





```

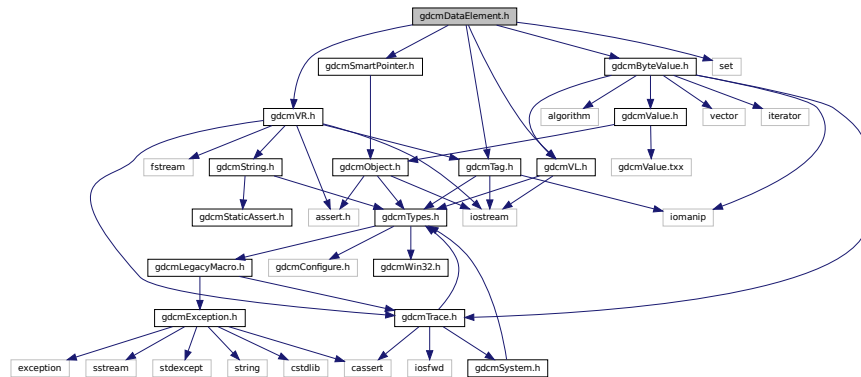
graph TD
    gdcmCurve.h --> gdcmObject.h
    gdcmCurve.h --> vector
    gdcmObject.h --> gdcmTypes.h
    gdcmObject.h --> assert.h
    gdcmObject.h --> iostream
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmException.h
    gdcmTypes.h --> gdcmTrace.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmException.h --> cassert
    gdcmException.h --> iosfwd
    gdcmException.h --> gdcmSystem.h
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h
    gdcmSystem.h --> gdcmCurve.h
  
```

- class `gdc::Curve`  
*Curve* class to handle element 50xx,3000 *Curve* Data.

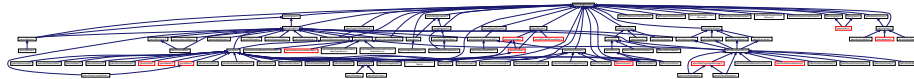
- **gdcm**

```
#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

- gdc

## Functions

- bool `gdcm::operator!=` (const DataElement &lhs, const DataElement &rhs)
- std::ostream & `gdcm::operator<<` (std::ostream &os, const DataElement &val)

## 11.54 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.*

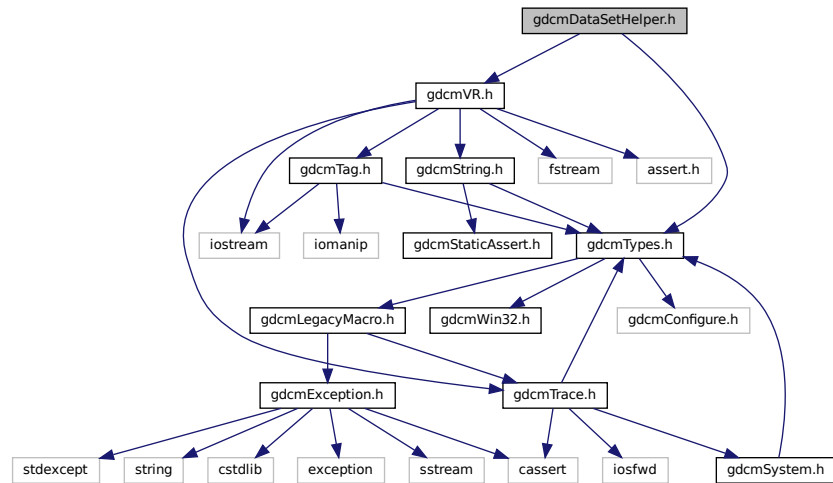
### Namespaces

- [gdcm](#)





Include dependency graph for `gdcmDataSetHelper.h`:



## Classes

- class [gdcm::DataSetHelper](#)  
*DataSetHelper* (internal class, not intended for user level)

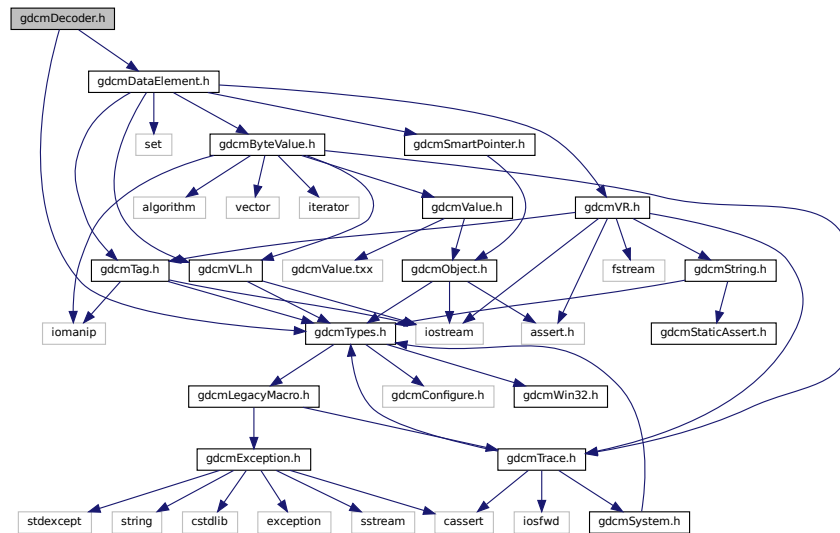
## Namespaces

- [gdcm](#)

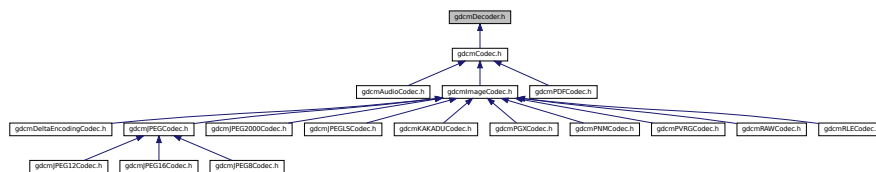
## 11.58 gdcmDecoder.h File Reference

```
#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.*

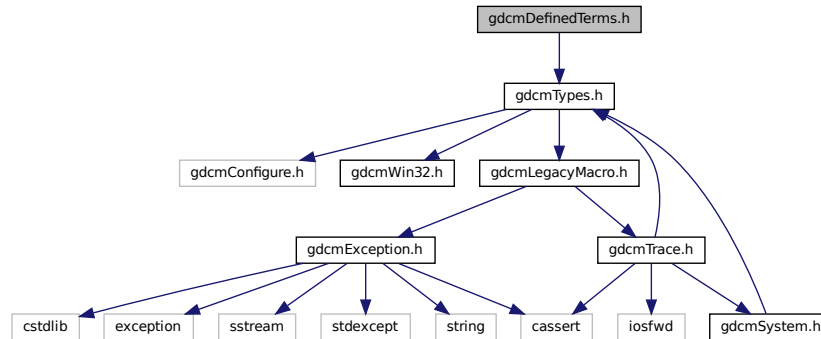
## Namespaces

- [gdcm](#)

## 11.59 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.*

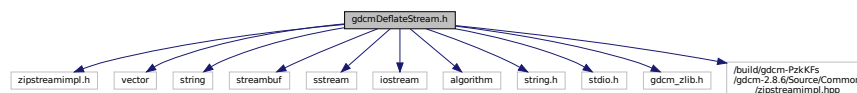
### Namespaces

- [gdcm](#)

## 11.60 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:

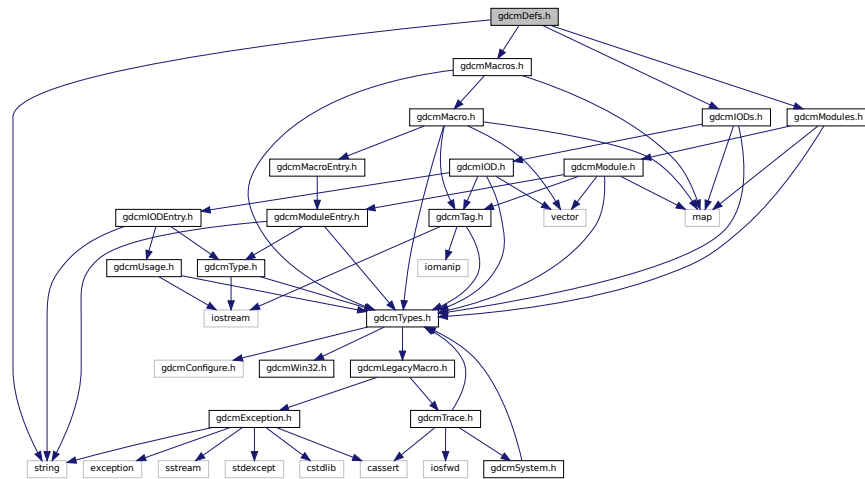




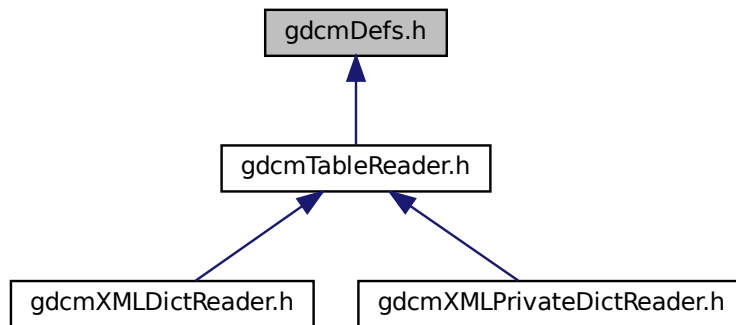
## 11.61 gdcmDefs.h File Reference

```
#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"
#include <string>
```

Include dependency graph for gdcmDefs.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::Defs](#)

*FIXME I do not like the name 'Defs'.*



[illegible]

- class `gdcm::DICOMDIR`  
*DICOMDIR* class.

- **gdcm**

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

```

graph TD
    Root[gdcmDICOMDIRGenerator.h] --> Tag[gdcmTag.h]
    Root --> Dir[gdcmDirectory.h]
    Root --> Util[utility]
    Tag --> IOmanip[iomanip]
    Tag --> Types[gdcmTypes.h]
    Dir --> Types
    Dir --> IOstream[iostream]
    Dir --> Vector[vector]
    Dir --> Assert[assert.h]
    Types --> LegacyMacro[gdcmLegacyMacro.h]
    Types --> Configure[gdcmConfigure.h]
    Types --> Win32[gdcmWin32.h]
    LegacyMacro --> Trace[gdcmTrace.h]
    LegacyMacro --> Exception[gdcmException.h]
    Trace --> System[gdcmSystem.h]
    Trace --> IOSfwd[iosfwd]
    Trace --> Cassert[cassert]
    Trace --> Sstream[sstream]
    Trace --> Stdexcept[stdexcept]
    Trace --> Cstdlib[cstdlib]
    Trace --> ExceptionLib[exception]
    Trace --> String[string]
    Exception --> IOSfwd
    Exception --> Cassert
    Exception --> Sstream
    Exception --> Stdexcept
    Exception --> Cstdlib
    Exception --> ExceptionLib
    Exception --> String
  
```



## Classes

- class [gdcm::Dict](#)  
Class to represent a map of [DictEntry](#).
- class [gdcm::PrivateDict](#)  
Private [Dict](#).

## Namespaces

- [gdcm](#)

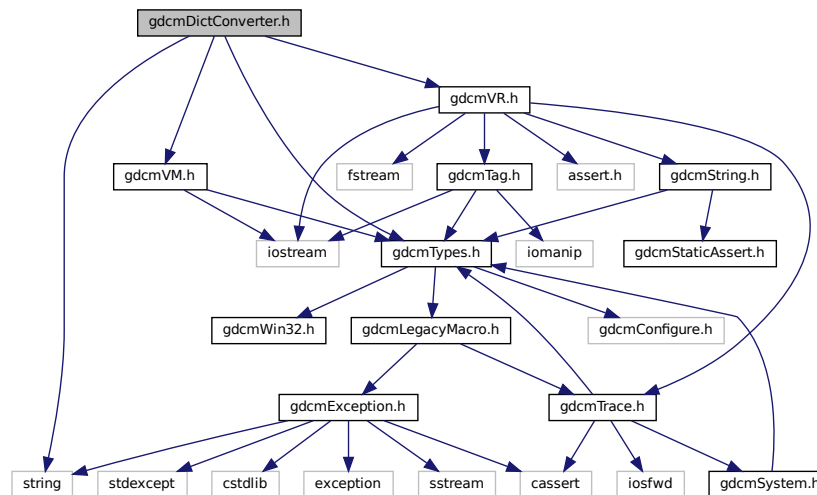
## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const Dict &val)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const PrivateDict &val)

## 11.66 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

- [gdcm](#)

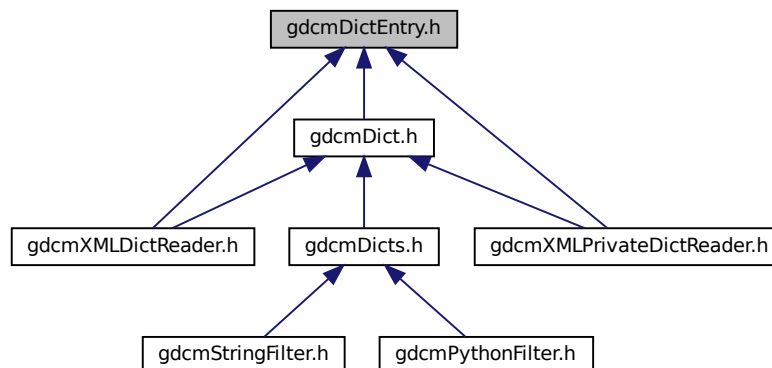
## 11.67 gdcmDictEntry.h File Reference

```
#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

- `gdcm`

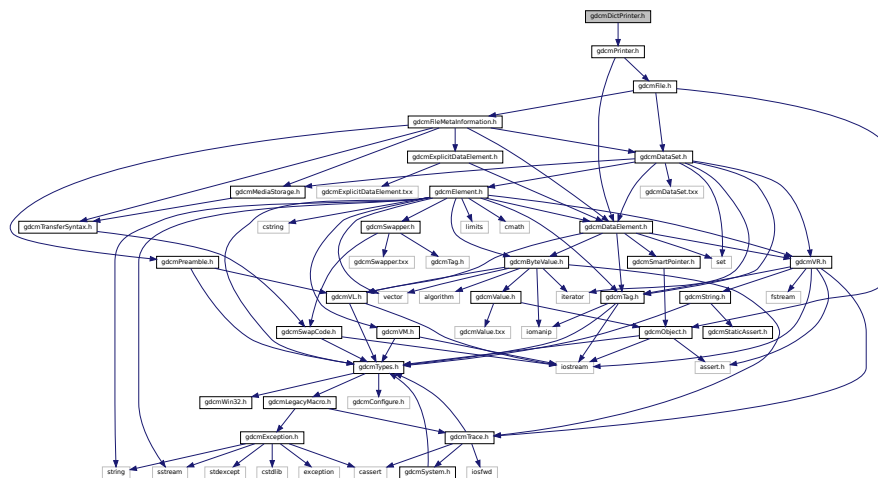
## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const DictEntry &val)`

## 11.68 gdcmDictPrinter.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcDictPrinter.h:



## Classes

- class `gdcm::DictPrinter`  
*DictPrinter* class.

## Namespaces

- **gdcm**





## 11.70 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

- `gdcm`
- `gdcm::network`

## 11.71 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmDirectionCosines.h`:



## Classes

- class [gdcm::DirectionCosines](#)  
class to handle *DirectionCosines*

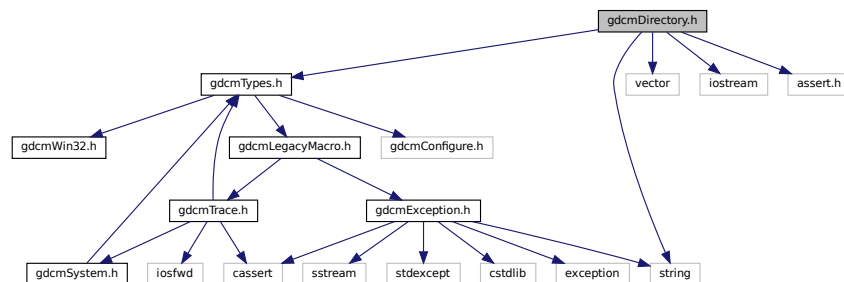
## Namespaces

- [gdcm](#)

## 11.72 gdcmDirectory.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
#include <assert.h>
```

Include dependency graph for `gdcmDirectory.h`:





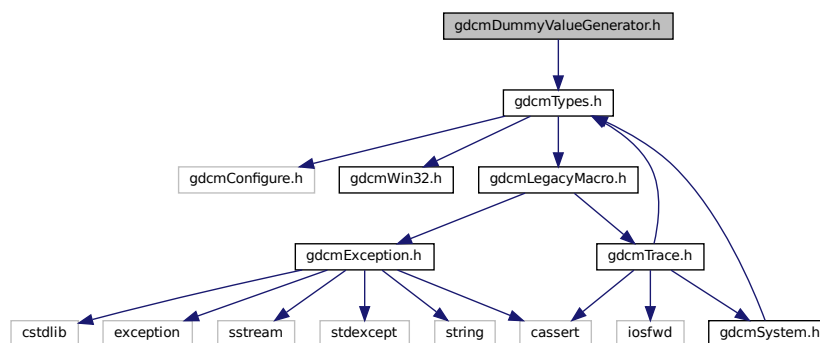
## Namespaces

- [gdcm](#)

## 11.74 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



## Classes

- class [gdcm::DummyValueGenerator](#)  
*Class for generating dummy value.*

## Namespaces

- [gdcm](#)

## 11.75 gdcmDumper.h File Reference

```
#include "gdcmPrinter.h"
```

[illegible]

- class `gdcm::Dumper`  
*Codec class.*

- **gdcm**

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmByteValue.h"
#include "gdcmDataElement.h"
#include "gdcmSwapper.h"
#include <string>
#include <vector>
#include <sstream>
#include <limits>
#include <cmath>
```



## Macros

- #define [VRDS16ILLEGAL](#)

## Functions

- ignore\_char const [gdcm::backslash](#) ('\\')
- std::istream & [gdcm::operator>>](#) (std::istream &in, ignore\_char const &ic)
- template<typename Float >  
std::string [gdcm::to\\_string](#) (Float data)

### 11.76.1 Macro Definition Documentation

#### 11.76.1.1 VRDS16ILLEGAL

```
#define VRDS16ILLEGAL
```

## 11.77 gdcmEmptyMaskGenerator.h File Reference

```
#include "gdcmSubject.h"
```

Include dependency graph for gdcmEmptyMaskGenerator.h:



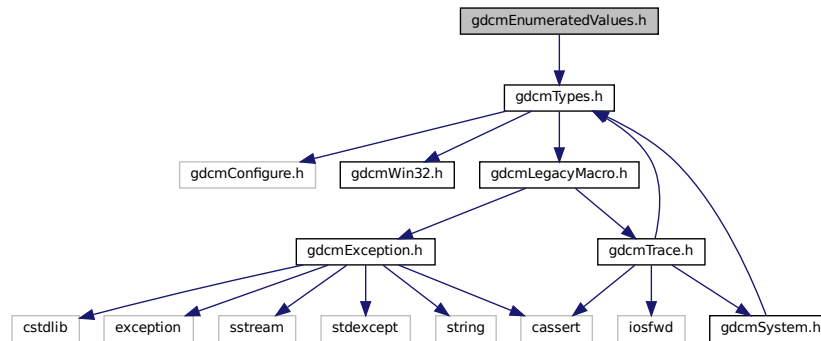




## 11.79 gdcmEnumeratedValues.h File Reference

```
#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

- [gdcm](#)

## 11.80 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 [gdc::AbortEvent](#)
- class [gdc::AnyEvent](#)
- class [gdc::EndEvent](#)
- class [gdc::Event](#)  
*superclass for callback/observer methods*
- class [gdc::ExitEvent](#)
- class [gdc::InitializeEvent](#)
- class [gdc::IterationEvent](#)
- class [gdc::ModifiedEvent](#)
- class [gdc::NoEvent](#)
- class [gdc::StartEvent](#)
- class [gdc::UserEvent](#)

## Namespaces

- [gdc](#)

## Macros

- `#define gdcEventMacro(classname, super)`

## Functions

- `std::ostream & gdc::operator<< (std::ostream &os, Event &e)`  
*Generic inserter operator for [Event](#) and its subclasses.*

## 11.80.1 Macro Definition Documentation

## 11.80.1.1 gdcmEventMacro

```
#define gdcmEventMacro(
    classname,
    super )
```

**Value:**

```
\
class classname : public super { \
public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() {} \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
    { return new Self; } \
    classname(const Self&s) : super(s){}; \
private: \
    void operator=(const Self&); \
}
```

## 11.81 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

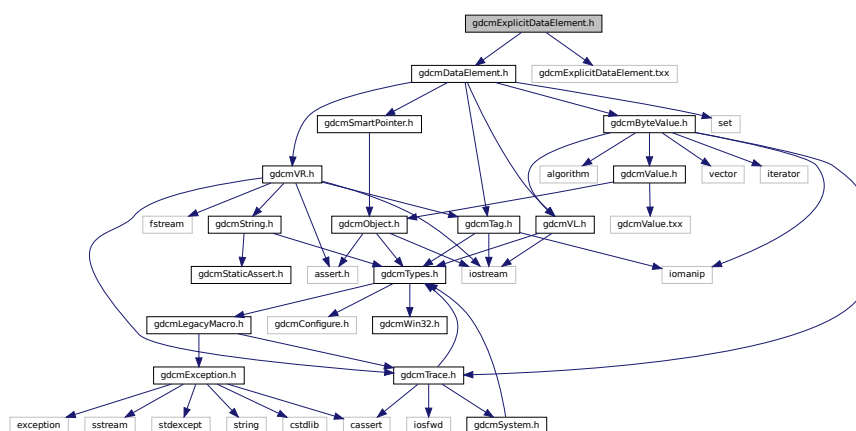
- [gdcm](#)

## 11.82 gdcmExplicitDataElement.h File Reference

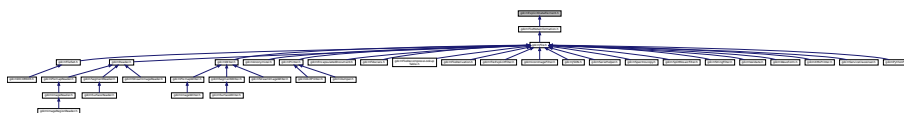
```
#include "gdcmDataElement.h"
```

```
#include "gdcmExplicitDataElement.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

- [gdcm](#)





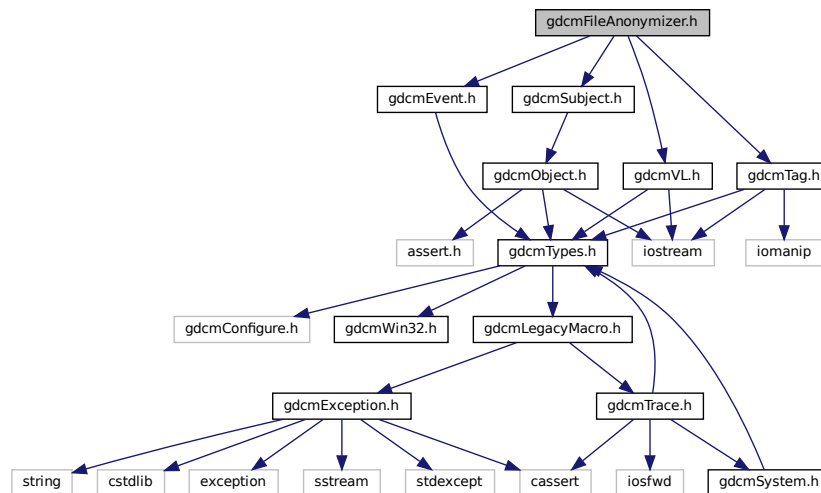
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const File &val)`

## 11.86 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

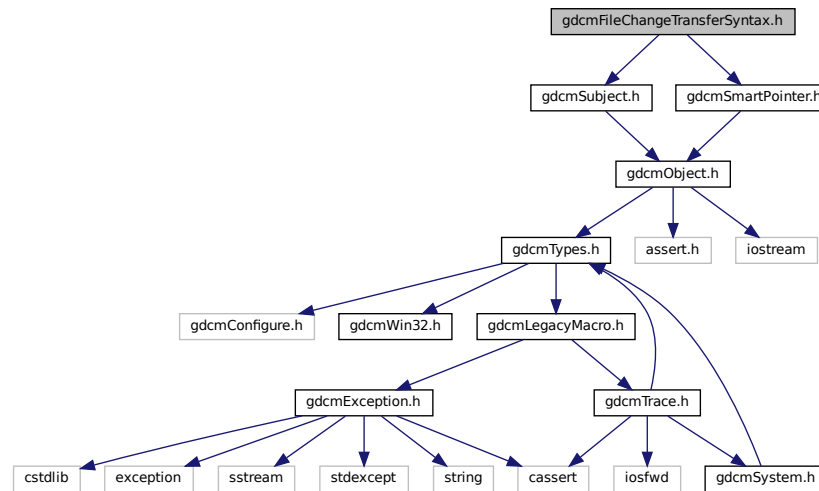
- `gdcm`

## 11.87 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



### Classes

- class [gdcm::FileChangeTransferSyntax](#)  
*FileChangeTransferSyntax.*

### Namespaces

- [gdcm](#)

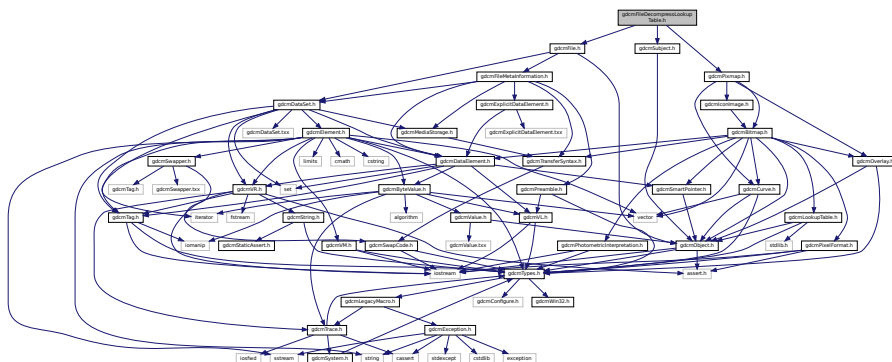
## 11.88 gdcmFileDecompressLookupTable.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmFile.h"
```



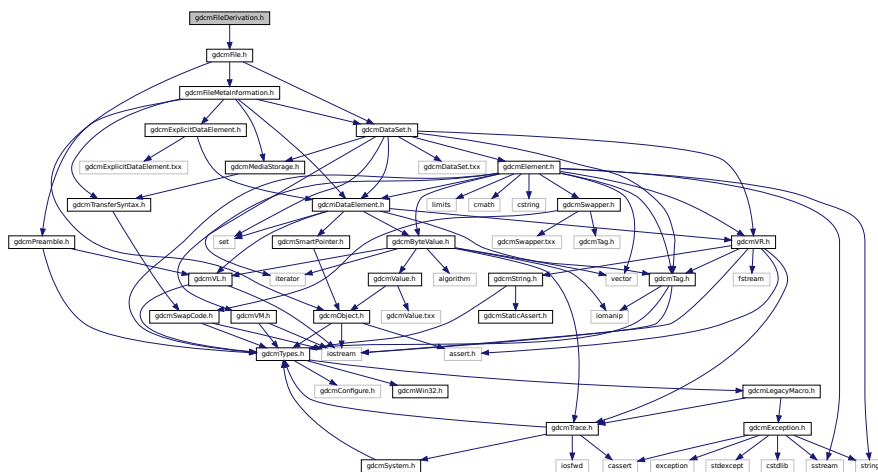
Include dependency graph for `gdcmFileDecompressLookupTable.h`:



- class `gdcm::FileDecompressLookupTable`  
*FileDecompressLookupTable* class.

- **gdcm**

Include dependency graph for `gdcmFileDerivation.h`:







## 11.92 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.*

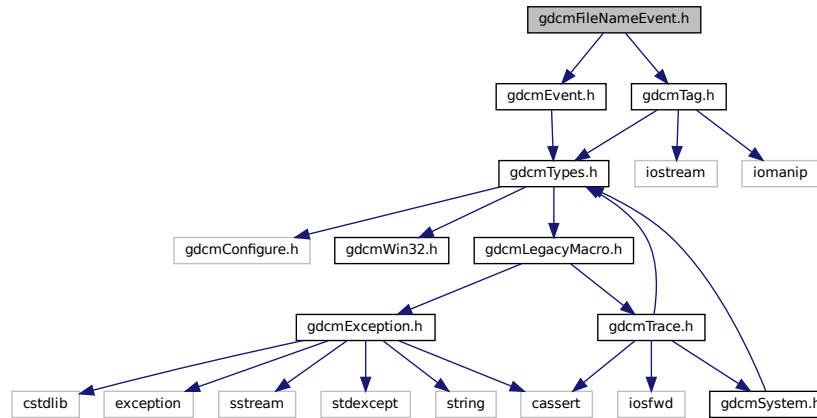
### Namespaces

- [gdcm](#)

## 11.93 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



## Classes

- class [gdcm::FileNameEvent](#)  
*FileNameEvent.*

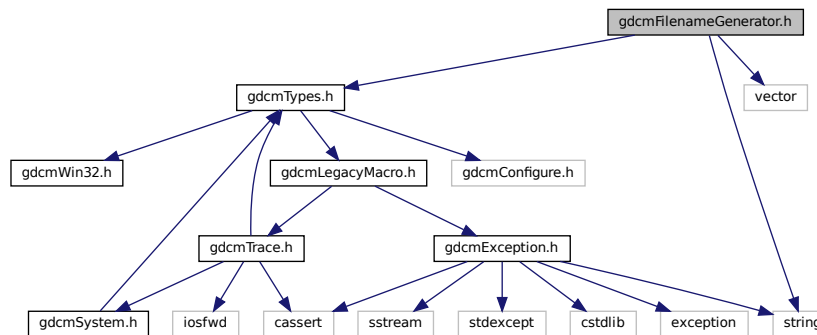
## Namespaces

- [gdcm](#)

## 11.94 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:





## Classes

- class [gdcm::FileSet](#)

## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)`

## 11.96 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
```

```
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



## Classes

- class [gdcm::FileStreamer](#)  
*FileStreamer.*

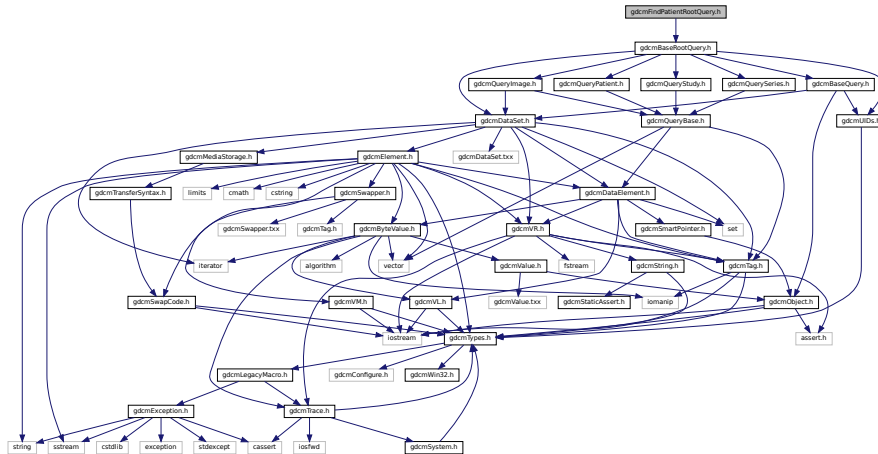
## Namespaces

- [gdcm](#)

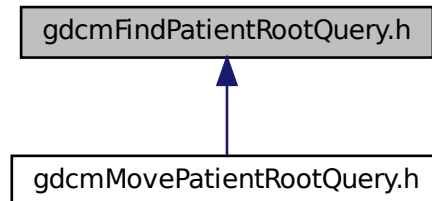
## 11.97 gdcmFindPatientRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindPatientRootQuery.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::FindPatientRootQuery](#)  
*PatientRootQuery.*

### Namespaces

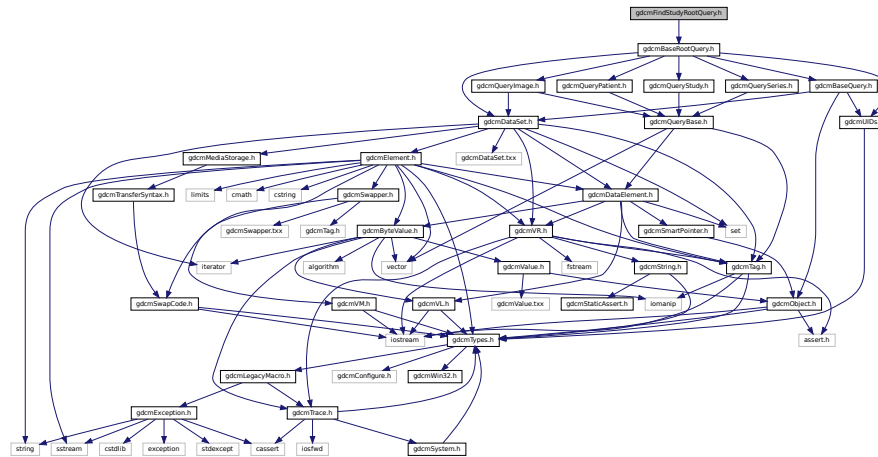
- [gdcm](#)



## 11.98 gdcmFindStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindStudyRootQuery.h:



### Classes

- class [gdcm::FindStudyRootQuery](#)  
*FindStudyRootQuery.*

### Namespaces

- [gdcm](#)

## 11.99 gdcmFragment.h File Reference

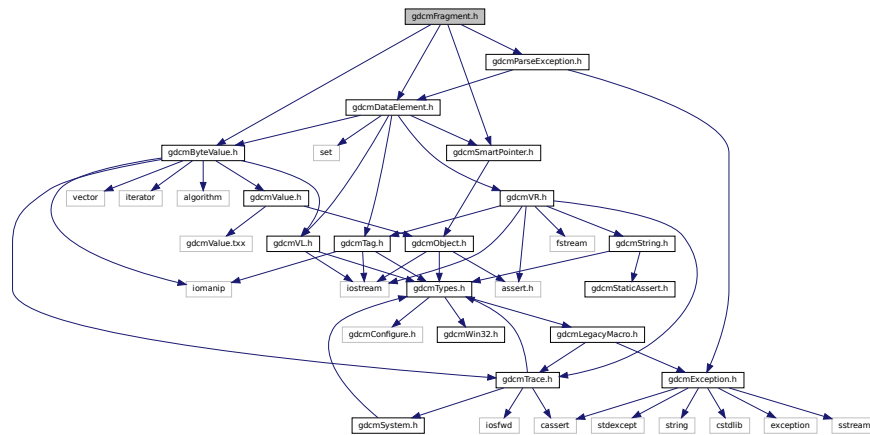
```
#include "gdcmDataElement.h"
```

```
#include "gdcmByteValue.h"
```

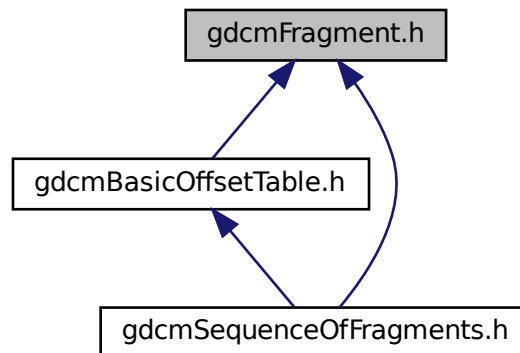
```
#include "gdcmSmartPointer.h"
```

```
#include "gdcmParseException.h"
```

Include dependency graph for `gdcmFragment.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Fragment`  
Class to represent a *Fragment*.

## Namespaces

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)`

## 11.100 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



## Classes

- class `gdcm::Global`  
*Global.*

## Namespaces

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)`

## Variables

- static Global `gdcm::GlobalInstance`

## 11.101 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

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)`

## 11.102 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

[illegible]

- **gdcm**

- typedef Bitmap **gdcm::IconImage**

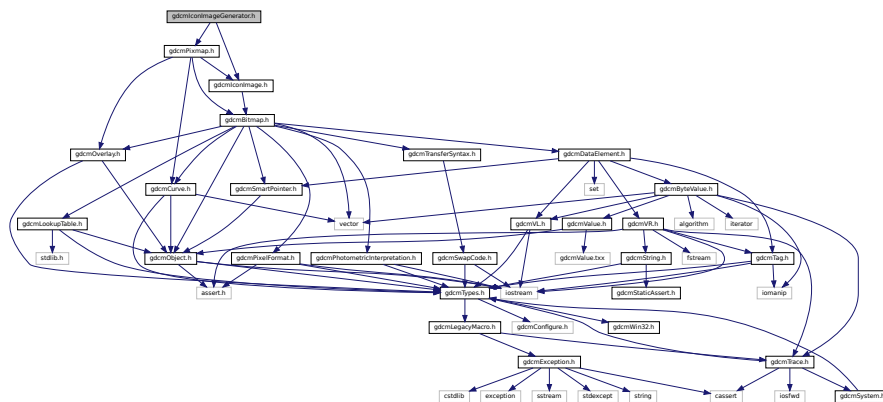
```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

[illegible]

- class `gdcm::IconImageFilter`  
*IconImageFilter*.

- `gdcm`

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
Include dependency graph for gdcmIconImageGenerator.h:
```





## Classes

- class [gdcm::Image](#)  
*Image.*

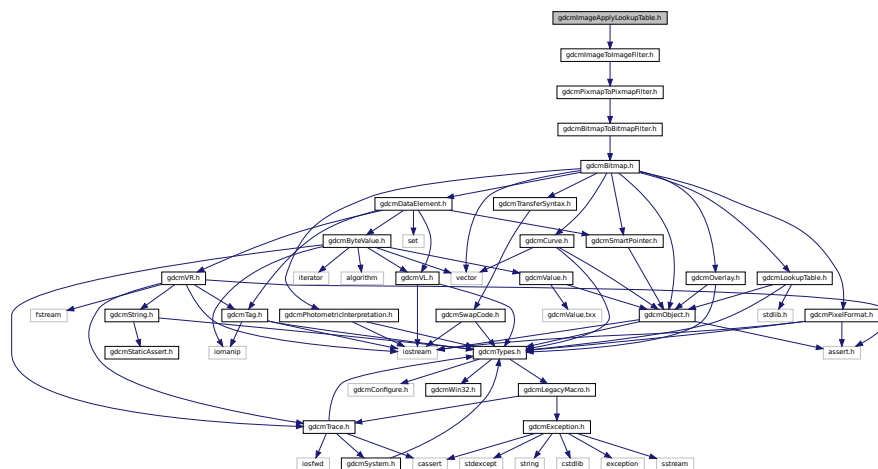
## Namespaces

- [gdcm](#)

## 11.106 gdcmImageApplyLookupTable.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageApplyLookupTable.h:



## Classes

- class [gdcm::ImageApplyLookupTable](#)  
*ImageApplyLookupTable class.*

## Namespaces

- [gdcm](#)

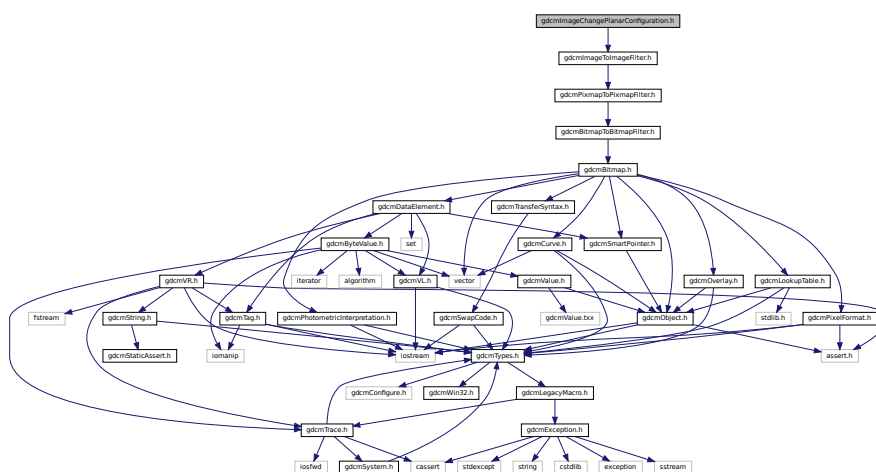




## 11.108 gdcmlImageChangePlanarConfiguration.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for `gdcmImageChangePlanarConfiguration.h`:



## Classes

- class `gdcm::ImageChangePlanarConfiguration`  
*ImageChangePlanarConfiguration* class.

## Namespaces

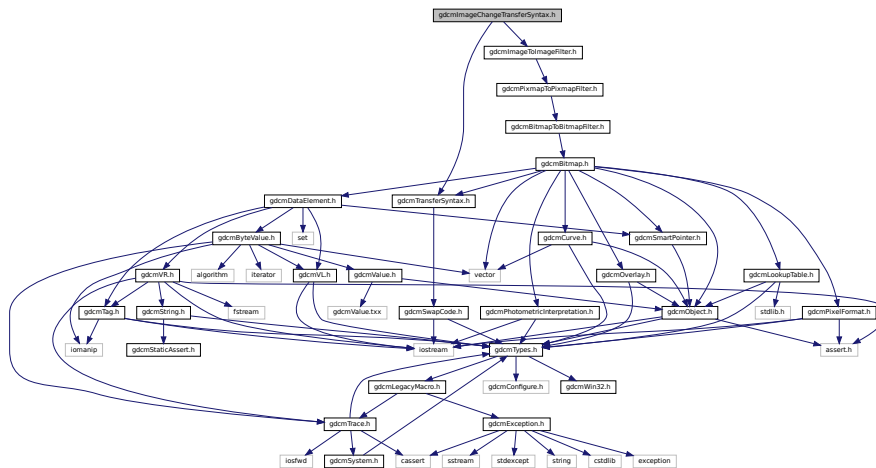
- **gdcm**

## 11.109 gdcmlImageChangeTransferSyntax.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

```
#include "gdcmTransferSyntax.h"
```

Include dependency graph for gdcmImageChangeTransferSyntax.h:



## Classes

- class [gdcm::ImageChangeTransferSyntax](#)  
*ImageChangeTransferSyntax* class.

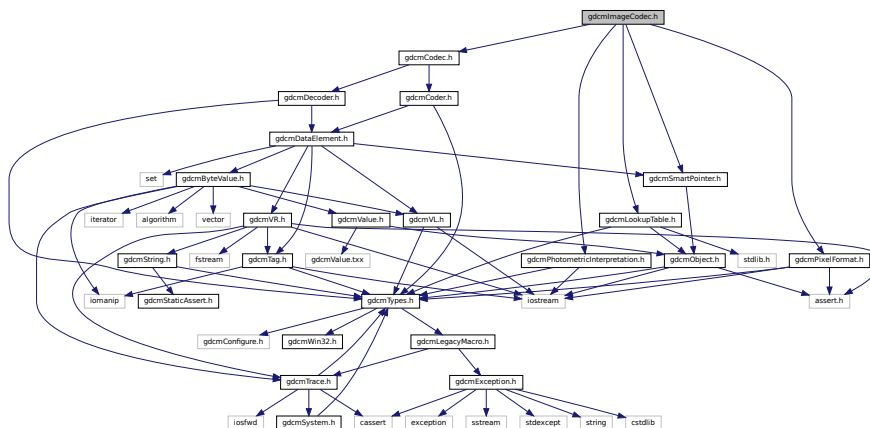
## Namespaces

- [gdcm](#)

## 11.110 gdcmImageCodec.h File Reference

```
#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for `gdcmImageCodec.h`:



This graph shows which files directly or indirectly include this file:



## Classes

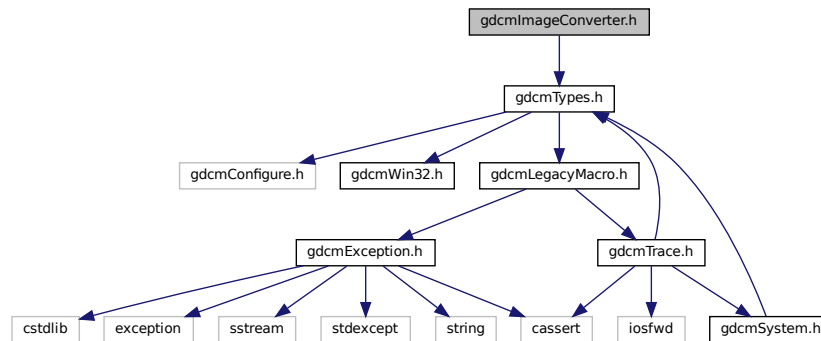
- class `gdcm::ImageCodec`  
*ImageCodec.*

## Namespaces

- `gdcm`

## 11.111 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

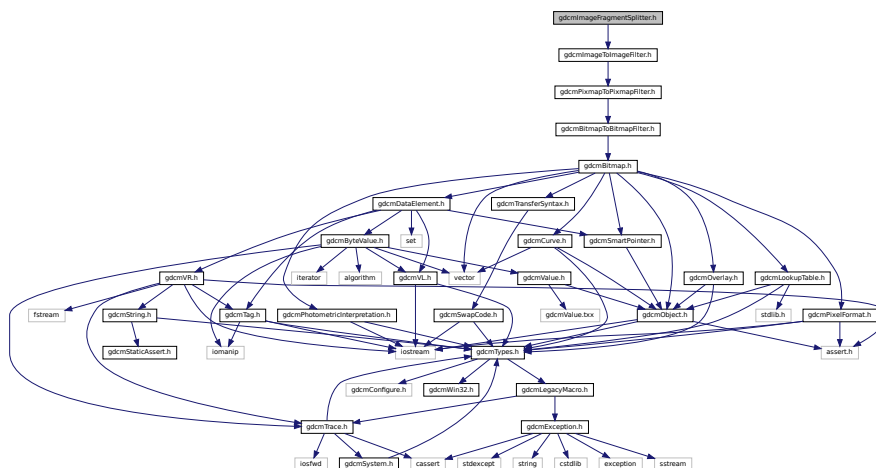


- class `gdcm::ImageConverter`  
*Image Converter*.

- **gdc**

Include dependency graph for `gdcmlImageFragmentSplitter.h`:

Include dependency graph for `gdcmlImageFragmentSplitter.h`:



## Classes

- class [gdcm::ImageFragmentSplitter](#)  
*ImageFragmentSplitter* class.

## Namespaces

- [gdcm](#)

### 11.113 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

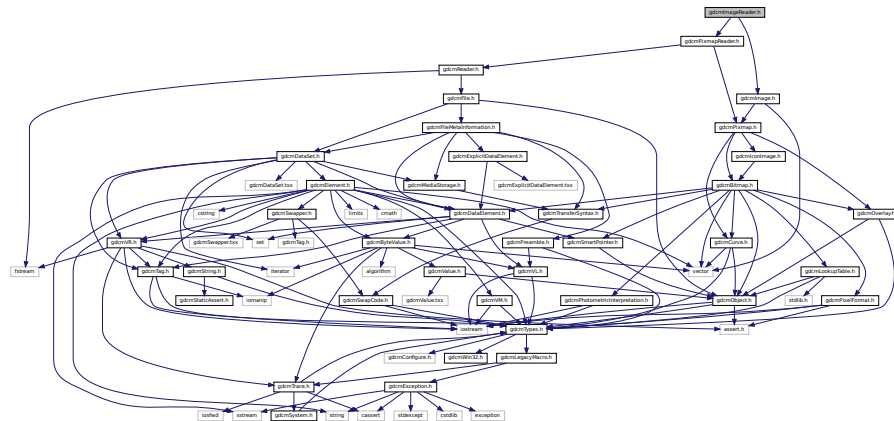
- class [gdcm::ImageHelper](#)  
*ImageHelper* (internal class, not intended for user level)
- struct [gdcm::RealWorldValueMappingContent](#)

## Namespaces

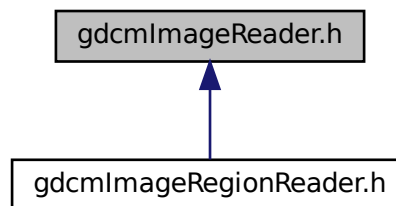
- [gdcm](#)

## 11.114 gdcmlImageReader.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 `gdcm::ImageReader`  
*ImageReader*.

## Namespaces

- **gdcm**

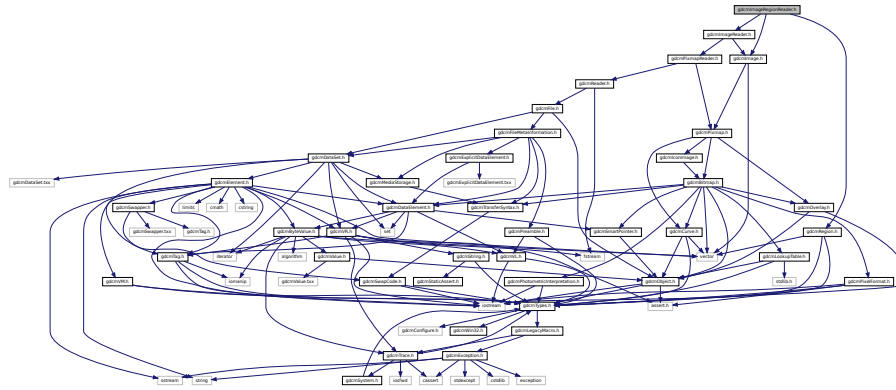
## 11.115 gdcmlImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
```

```
#include "gdcmImage.h"
```

```
#include "gdcmRegion.h"
```

Include dependency graph for `gdcmlImageRegionReader.h`:



## Classes

- class `gdcm::ImageRegionReader`  
*ImageRegionReader.*

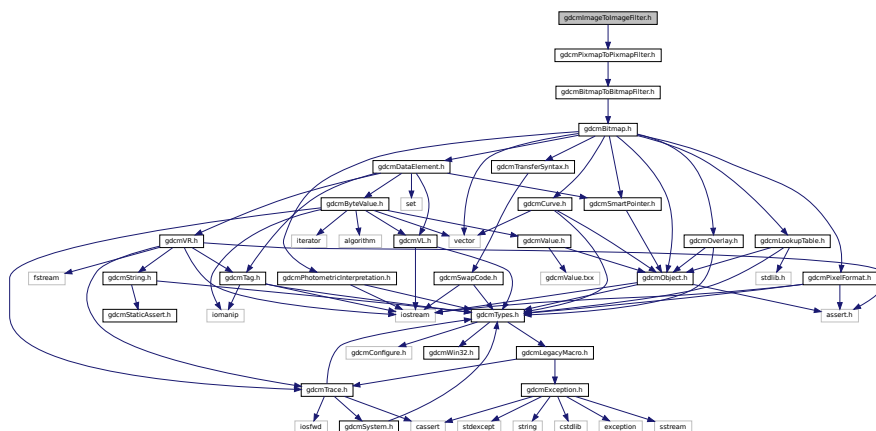
## Namespaces

- **gdcm**

## 11.116 gdcmlImageToImageFilter.h File Reference

```
#include "gdcmPixmapToPixmapFilter.h"
```

Include dependency graph for `gdcmImageToImageFilter.h`:





```

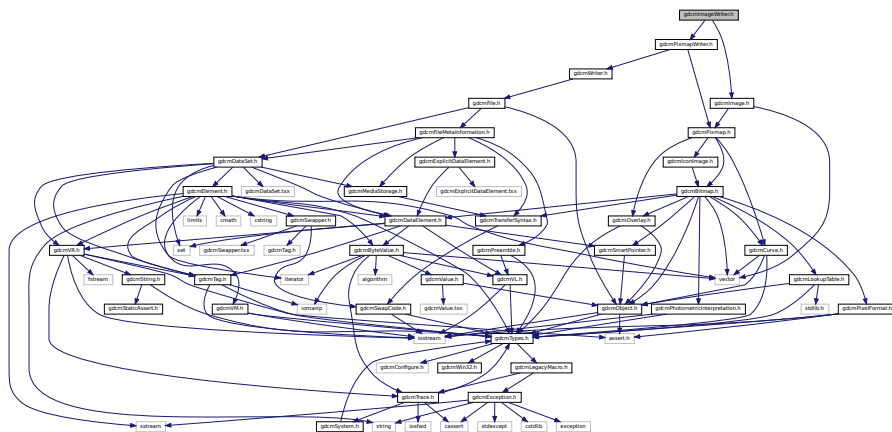
classDiagram
    class gdcmImageToImageFilter_h["gdcmImageToImageFilter.h"]
    class gdcmImageApplyLookupTable_h["gdcmImageApplyLookupTable.h"]
    class gdcmImageChangePhotometricInterpretation_h["gdcmImageChangePhotometricInterpretation.h"]
    class gdcmImageChangePlanarConfiguration_h["gdcmImageChangePlanarConfiguration.h"]
    class gdcmImageChangeTransferSyntax_h["gdcmImageChangeTransferSyntax.h"]
    class gdcmImageFragmentSplitter_h["gdcmImageFragmentSplitter.h"]

    gdcmImageApplyLookupTable_h --|> gdcmImageToImageFilter_h
    gdcmImageChangePhotometricInterpretation_h --|> gdcmImageToImageFilter_h
    gdcmImageChangePlanarConfiguration_h --|> gdcmImageToImageFilter_h
    gdcmImageChangeTransferSyntax_h --|> gdcmImageToImageFilter_h
    gdcmImageFragmentSplitter_h --|> gdcmImageToImageFilter_h
  
```

- class `gdcm::ImageToImageFilter`  
*ImageToImageFilter* class.

- **gdcm**

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
Include dependency graph for gdcmImageWriter.h:
```



- class `gdcm::ImageWriter`  
*ImageWriter*.

## Namespaces

- [gdcm](#)

### 11.118 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.*

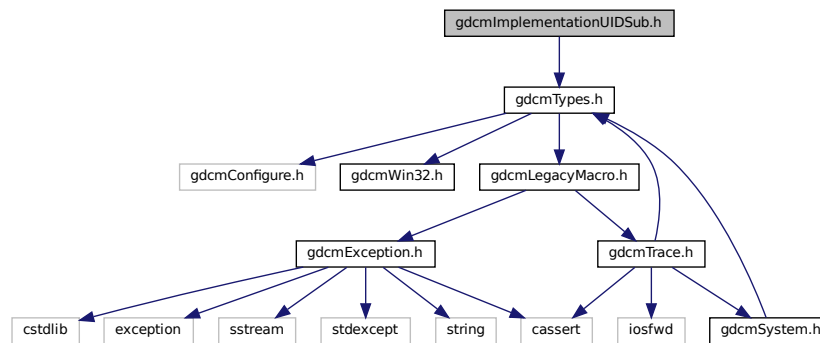
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.119 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



## Classes

- class [gdcm::network::ImplementationUIDSub](#)  
*ImplementationUIDSub.*

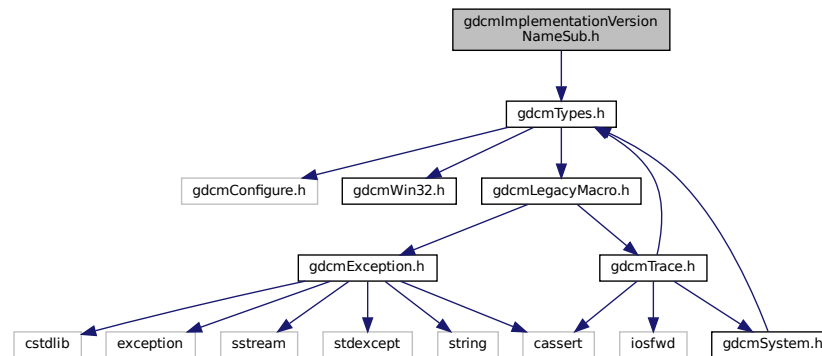
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

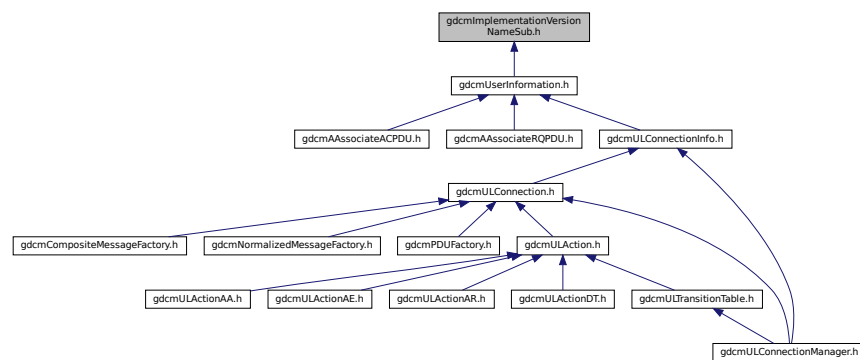
## 11.120 gdcmImplementationVersionNameSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmImplementationVersionNameSub.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::ImplementationVersionNameSub`  
*ImplementationVersionNameSub.*

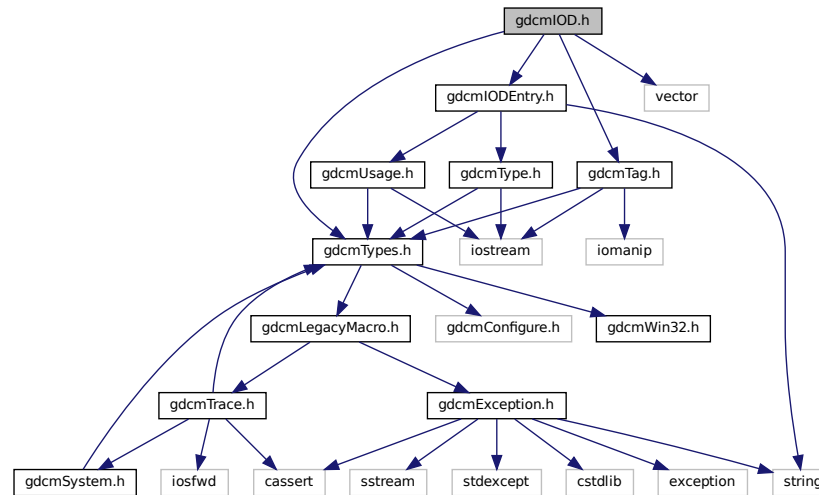
## Namespaces

- `gdcm`
- `gdcm::network`

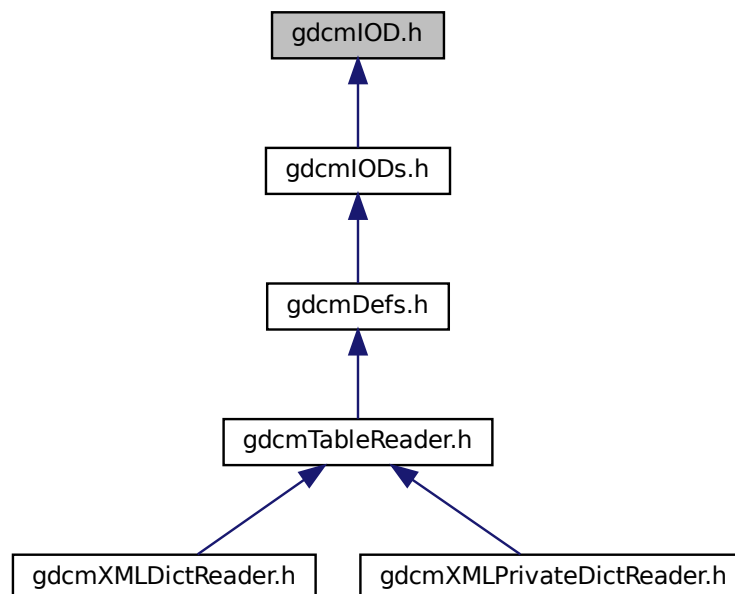


```
#include <vector>
```

Include dependency graph for gdcmlOD.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::IOD](#)

*Class for representing a [IOD](#).*

## Namespaces

- [gdcm](#)

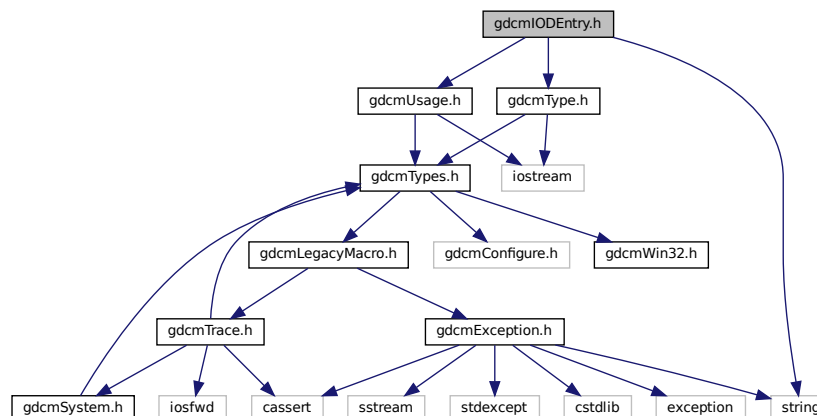
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

## 11.123 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 [gdcml::IODEntry](#)  
*Class for representing a [IODEntry](#).*

## Namespaces

- [gdcml](#)

## Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IODEntry &_val)`



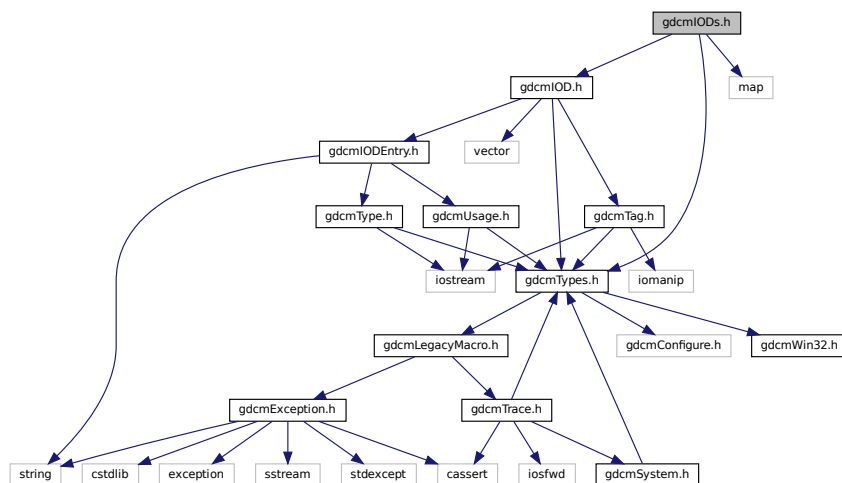
## 11.124 gdcmIODs.h File Reference

```
#include "gdcmTypes.h"
```

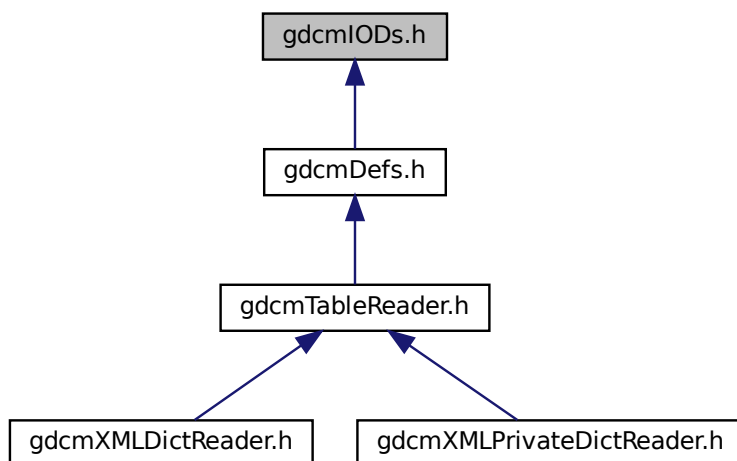
```
#include "gdcmIOD.h"
```

```
#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

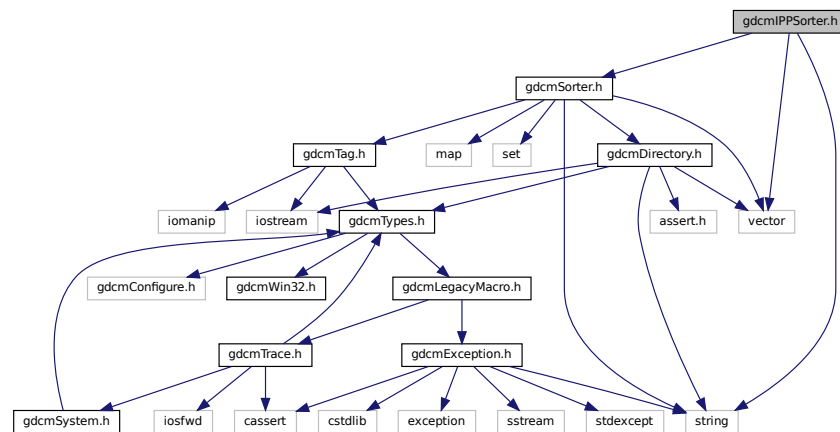
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)`

## 11.125 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
#include <vector>
#include <string>
Include dependency graph for gdcmIPPSorter.h:
```



## Classes

- class [gdcm::IPPSorter](#)  
*IPPSorter.*

## Namespaces

- [gdcm](#)

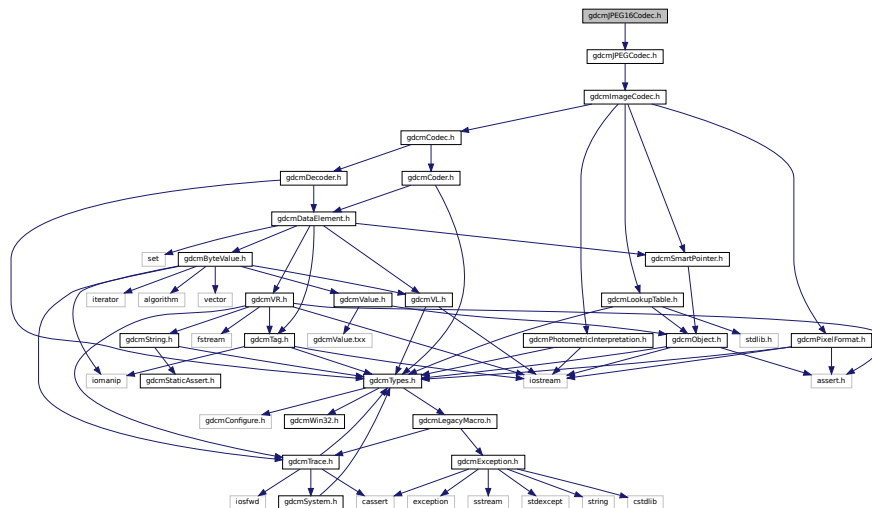




## 11.128 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG16Codec.h:



### Classes

- class [gdcm::JPEG16Codec](#)  
*Class to do JPEG 16bits (lossless)*

### Namespaces

- [gdcm](#)

## 11.129 gdcmJPEG2000Codec.h File Reference

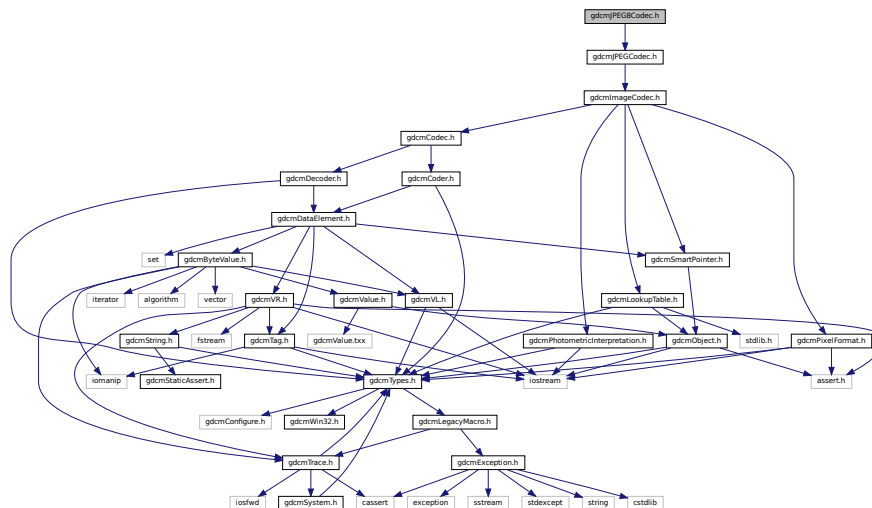
```
#include "gdcmImageCodec.h"
```



- class `gdcm::JPEG2000Codec`

- **gdcm**

Include dependency graph for gdcMJPEG8Codec.h:



## Classes

- class [gdcm::JPEG8Codec](#)  
*Class to do JPEG 8bits (lossy & lossless)*

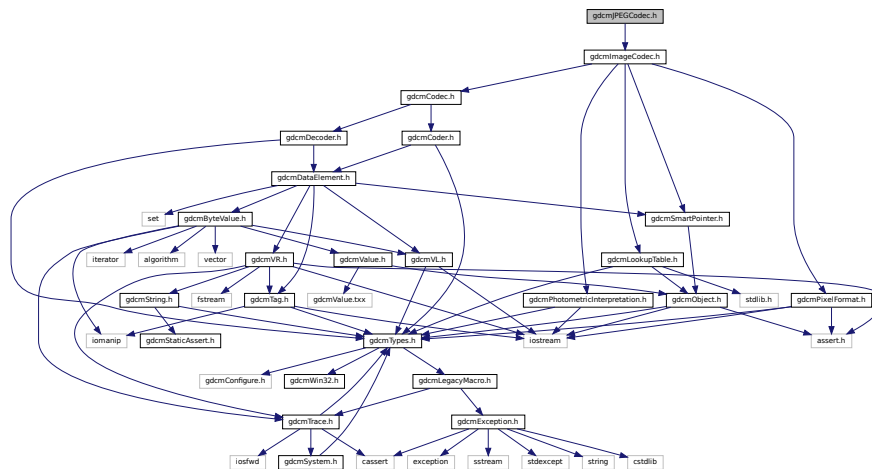
## Namespaces

- [gdcm](#)

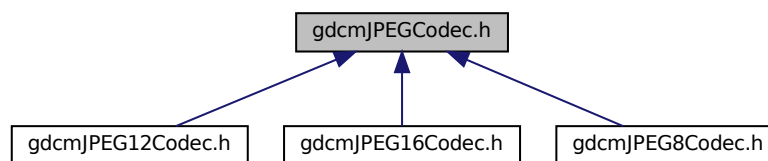
## 11.131 gdcmJPEGCodec.h File Reference

```
#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.*





[illegible]

- class `gdcm::JSON`

- gdcm

```
#include "gdcmImageCodec.h"
Include dependency graph for gdcmKAKADUCodec.h:
```



## Classes

- class [gdcm::KAKADUCodec](#)  
*KAKADUCodec.*

## Namespaces

- [gdcm](#)

## 11.135 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) method;`
- `#define GDCM_LEGACY_BODY(method, version) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`
- `#define GDCM_LEGACY_REPLACED_BODY(method, version, replace) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.)"`

### 11.135.1 Macro Definition Documentation

## 11.135.1.1 GDCM\_LEGACY

```
#define GDCM_LEGACY(  
    method ) method;
```

## 11.135.1.2 GDCM\_LEGACY\_BODY

```
#define GDCM_LEGACY_BODY(  
    method,  
    version ) gdcmWarningMacro(#method " was deprecated for " version " and will be  
removed in a future version.")
```

## 11.135.1.3 GDCM\_LEGACY\_REPLACED\_BODY

```
#define GDCM_LEGACY_REPLACED_BODY(  
    method,  
    version,  
    replace ) gdcmWarningMacro(#method " was deprecated for " version " and will be  
removed in a future version. Use " #replace " instead.)
```

## 11.136 gdcmLO.h File Reference

```
#include "gdcmString.h"  
Include dependency graph for gdcmLO.h:
```



## Classes

- class [gdcm::LO](#)  
[LO](#).

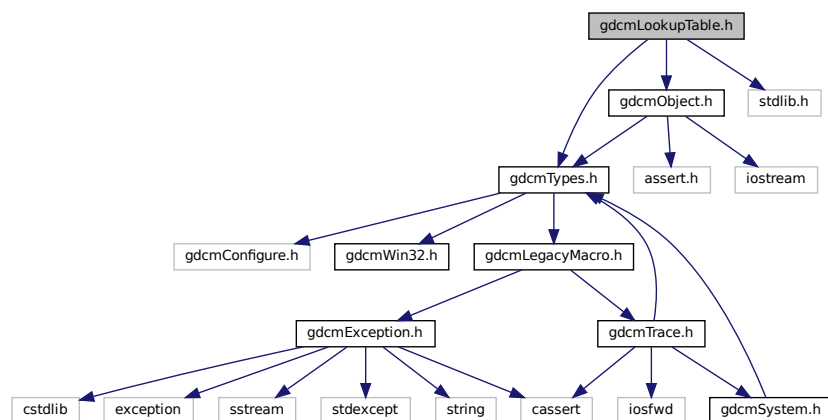
## Namespaces

- [gdcm](#)

## 11.137 gdcmLookupTable.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>
```

Include dependency graph for gdcmLookupTable.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::LookupTable](#)  
[LookupTable](#) class.

## Namespaces

- [gdcm](#)

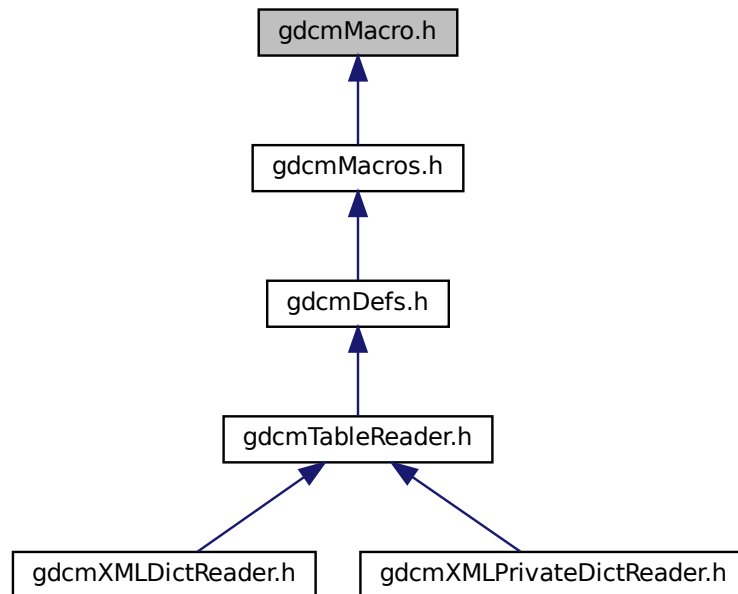
## 11.138 gdcmMacro.h File Reference

```
#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

- [gdc](#)

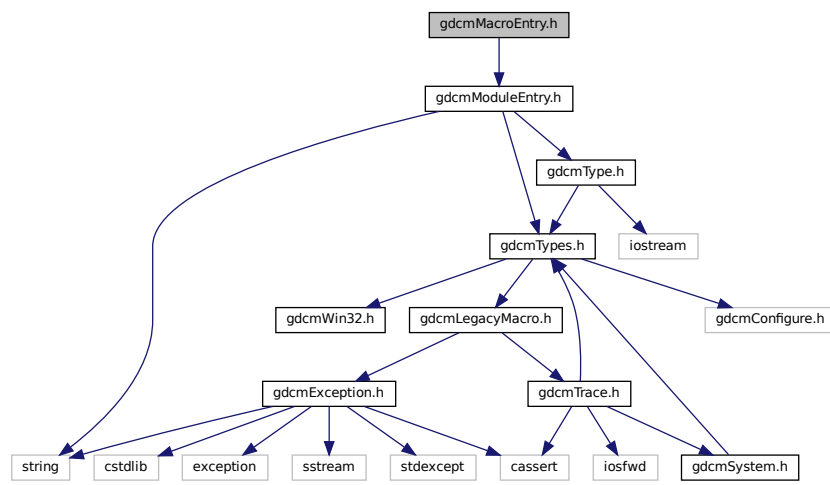
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

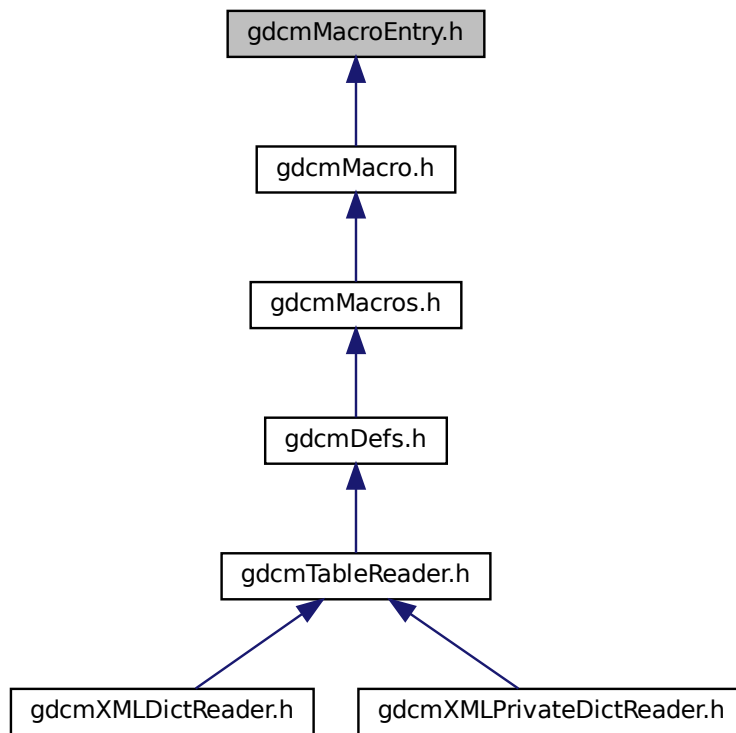
## 11.139 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define GDCMMACROENTRY_H`

### 11.139.1 Macro Definition Documentation

#### 11.139.1.1 GDCMMACROENTRY\_H

```
#define GDCMMACROENTRY_H
```



## 11.140 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 [gdc::Macros](#)  
*Class for representing a [Modules](#).*

## Namespaces

- [gdc](#)

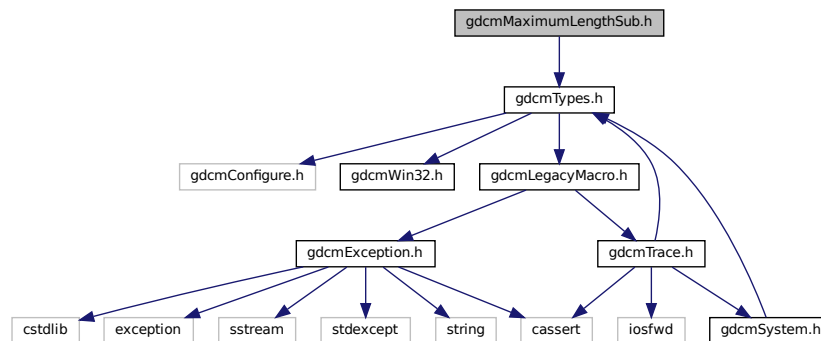
## Functions

- `std::ostream & gdc::operator<< (std::ostream &_os, const Macros &_val)`

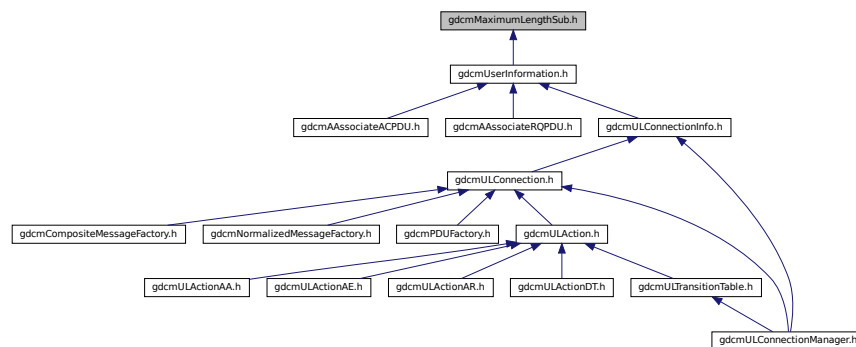
## 11.141 gdcMaximumLengthSub.h File Reference

```
#include "gdcTypes.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

- `gdcm`
- `gdcm::network`

## 11.142 gdcmMD5.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMD5.h:



## Classes

- class [gdcm::MD5](#)

*Class for [MD5](#).*

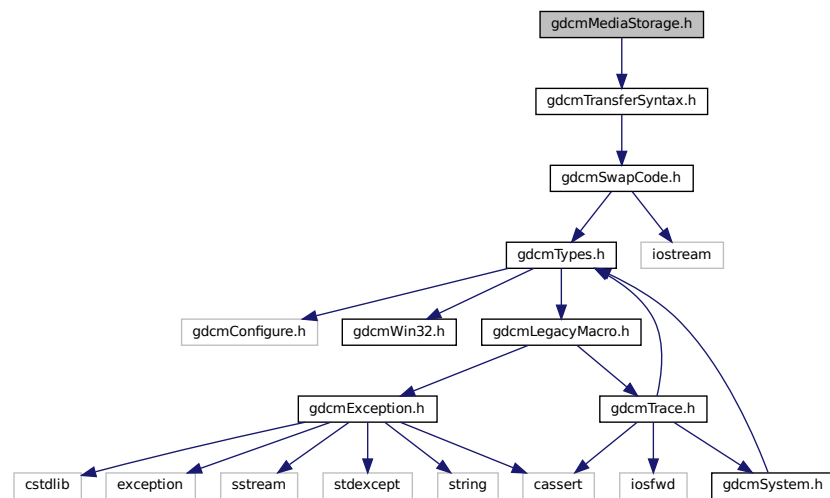
## Namespaces

- [gdcm](#)

## 11.143 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

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

## 11.144 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:









```
#include <vector>
```

Include dependency graph for gdcmModule.h:

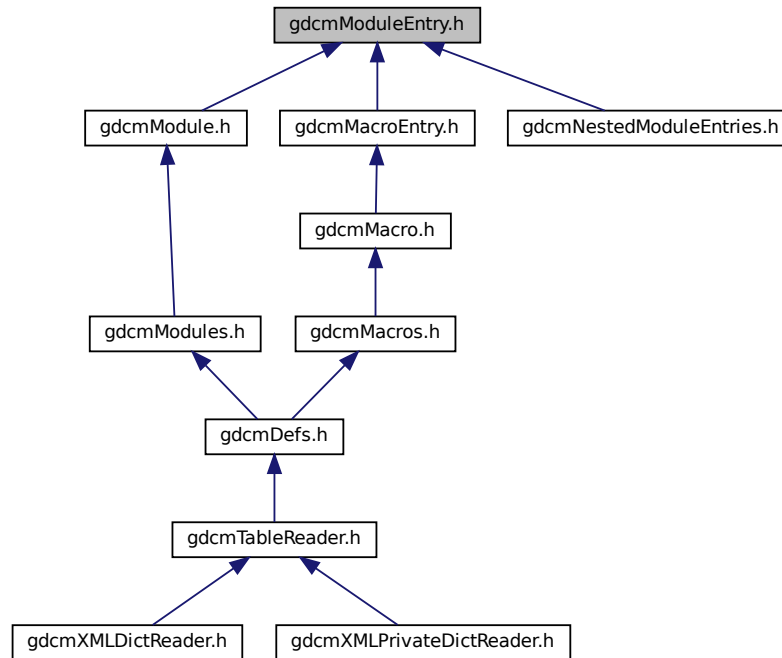


This graph shows which files directly or indirectly include this file:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::ModuleEntry](#)  
*Class for representing a [ModuleEntry](#).*

## Namespaces

- [gdcm](#)

## Typedefs

- typedef ModuleEntry [gdcm::MacroEntry](#)

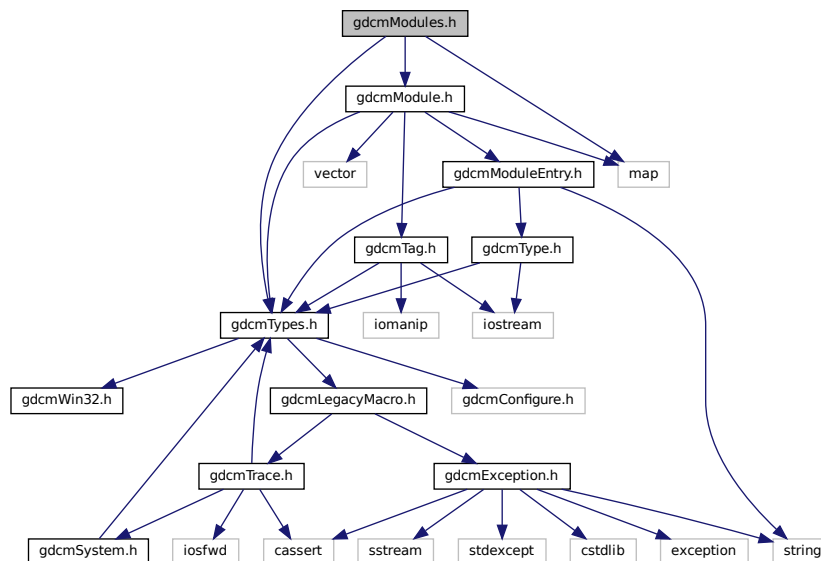
## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const ModuleEntry &\_val)

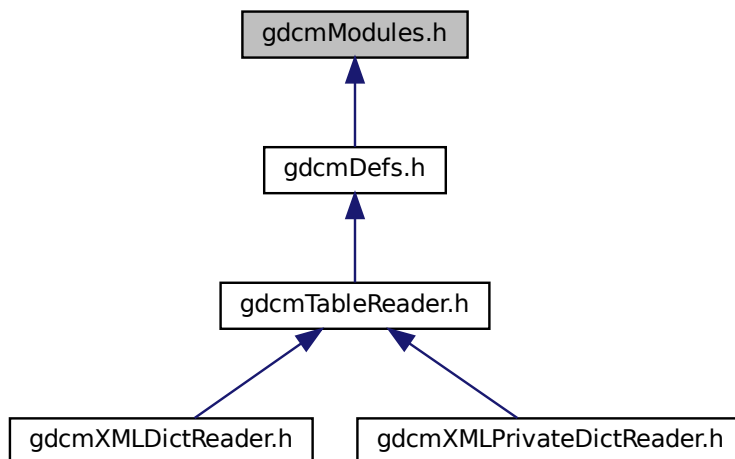
## 11.149 gdcmModules.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmModule.h"
#include <map>
```

Include dependency graph for gdcmModules.h:



This graph shows which files directly or indirectly include this file:

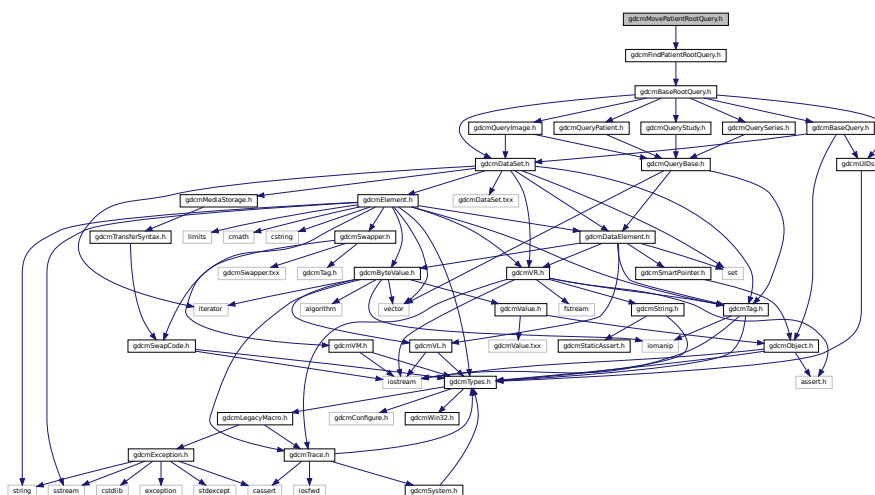


- class `gdcm::Modules`  
*Class for representing a `Modules`.*

- **gdcm**

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

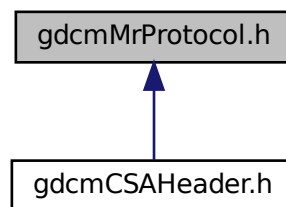
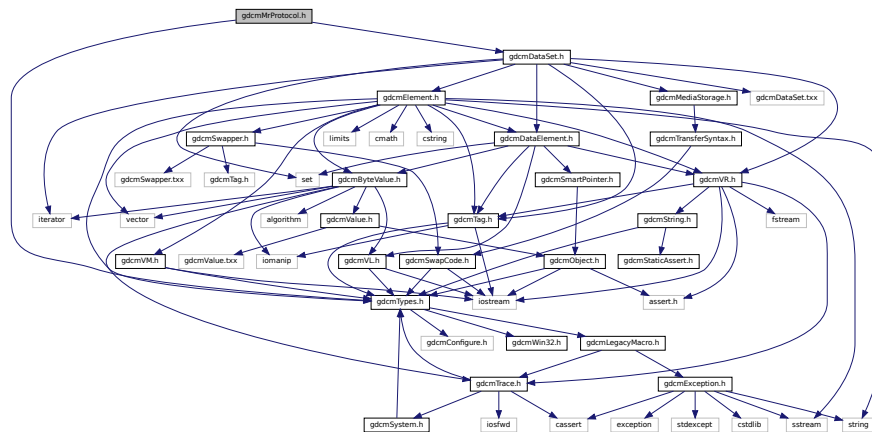
```
#include "gdcmFindPatientRootQuery.h"
Include dependency graph for gdcmMovePatientRootQuery.h:
```



- class `gdcm::MovePatientRootQuery`  
*MovePatientRootQuery.*

- **gdcm**





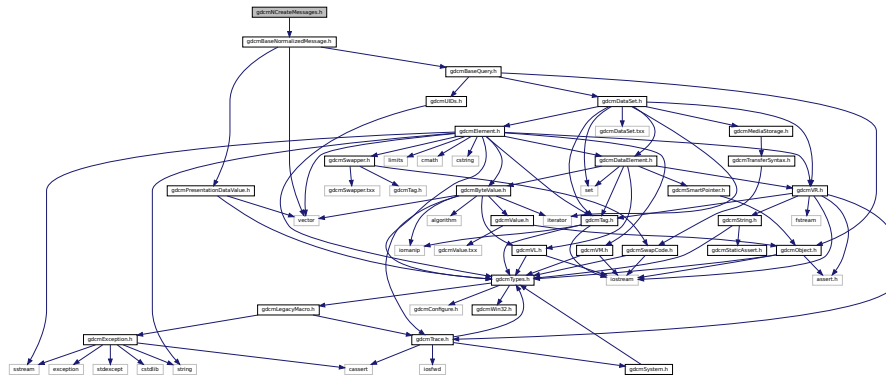
- class `gdcmm::MrProtocol`  
*Class for `MrProtocol`.*
- struct `gdcmm::MrProtocol::Slice`
- struct `gdcmm::MrProtocol::SliceArray`
- struct `gdcmm::MrProtocol::Vector3`

- **gdcm**





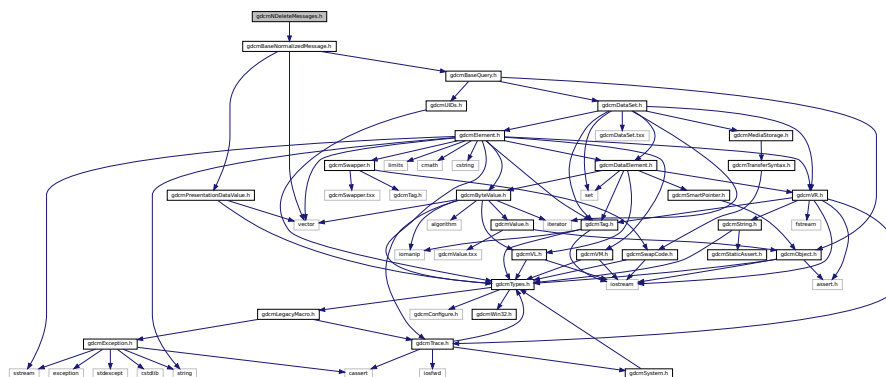
```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNCreateMessages.h:
```



- class `gdcmm::network::NCreateRQ`  
`NCreateRQ`.
- class `gdcmm::network::NCreateRSP`  
`NCreateRSP` this file defines the messages for the ncreate action.

- `gdcm`
- `gdcm::network`

```
#include "gdcmBaseNormalizedMessage.h"
Include dependency graph for gdcmNDeleteMessages.h:
```



## Classes

- class [gdcm::network::NDeleteRQ](#)  
*NDeleteRQ.*
- class [gdcm::network::NDeleteRSP](#)  
*NDeleteRSP* this file defines the messages for the *ndelete* action.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.156 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

- [gdcm](#)

## Typedefs

- typedef NestedModuleEntries [gdcm::NestedMacroEntries](#)

## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const NestedModuleEntries &\_val)

## 11.157 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- [gdcm](#)
- [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](#),

```

gdcmm::network::eARELEASEResponse,
gdcmm::network::eAABORTRequest,
gdcmm::network::eAABORTPDUPReceivedOpen,
gdcmm::network::eTransportConnectionClosed,
gdcmm::network::eARTIMTimerExpired,
gdcmm::network::eUnrecognizedPDUPReceived,
gdcmm::network::eEventDoesNotExist }

```

## Variables

- const int `gdcmm::network::cMaxEventID` = eEventDoesNotExist

## 11.158 gdcmmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- `gdcmm`
- `gdcmm::network`

## Enumerations

- enum `gdcmm::network::EStateID` {  
`gdcmm::network::eStaDoesNotExist` = 0,  
`gdcmm::network::eSta1Idle` = 1,  
`gdcmm::network::eSta2Open` = 2,  
`gdcmm::network::eSta3WaitLocalAssoc` = 4,  
`gdcmm::network::eSta4LocalAssocDone` = 8,  
`gdcmm::network::eSta5WaitRemoteAssoc` = 16,  
`gdcmm::network::eSta6TransferReady` = 32,  
`gdcmm::network::eSta7WaitRelease` = 64,  
`gdcmm::network::eSta8WaitLocalRelease` = 128,  
`gdcmm::network::eSta9ReleaseCollisionRqLocal` = 256,  
`gdcmm::network::eSta10ReleaseCollisionAc` = 512,  
`gdcmm::network::eSta11ReleaseCollisionRq` = 1024,  
`gdcmm::network::eSta12ReleaseCollisionAcLocal` = 2048,  
`gdcmm::network::eSta13AwaitingClose` = 4096 }

- int `gdcn::network::GetStateIndex` (EStateID inState)

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



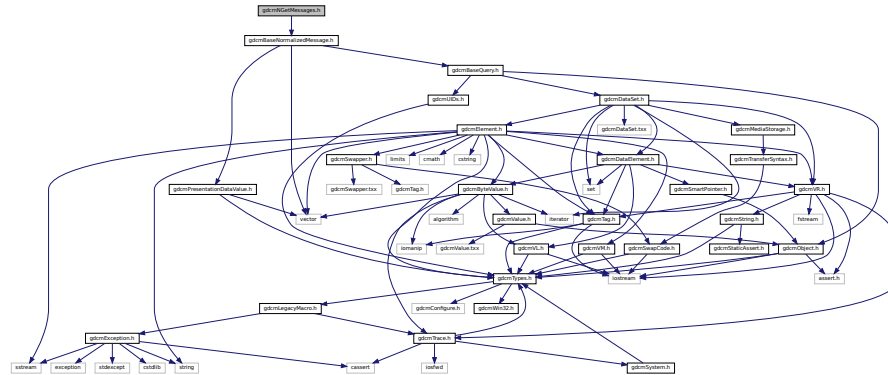
- class `gdcmm::network::NEventReportRQ`  
`NEventReportRQ`.
- class `gdcmm::network::NEventReportRSP`  
`NEventReportRSP` this file defines the messages for the neventreport action.

- `gdcm`
- `gdcm::network`

## 11.160 gdcMNGetMessages.h File Reference

```
#include "gdcmbaseNormalizedMessage.h"
```

Include dependency graph for gdcMNGetMessages.h:



## Classes

- class `gdcmm::network::NGetRQ`  
`NGetRQ`.
- class `gdcmm::network::NGetRSP`  
`NGetRSP` this file defines the messages for the `nget` action.

## Namespaces

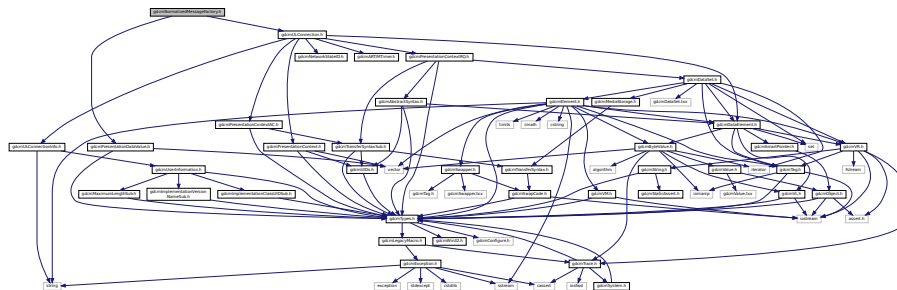
- `gdc`
- `gdc::network`

## 11.161 gdcmNormalizedMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
```

```
#include "gdcmULConnection.h"
```

Include dependency graph for `gdcmNormalizedMessageFactory.h`:



## Classes

- class `gdcm::network::NormalizedMessageFactory`

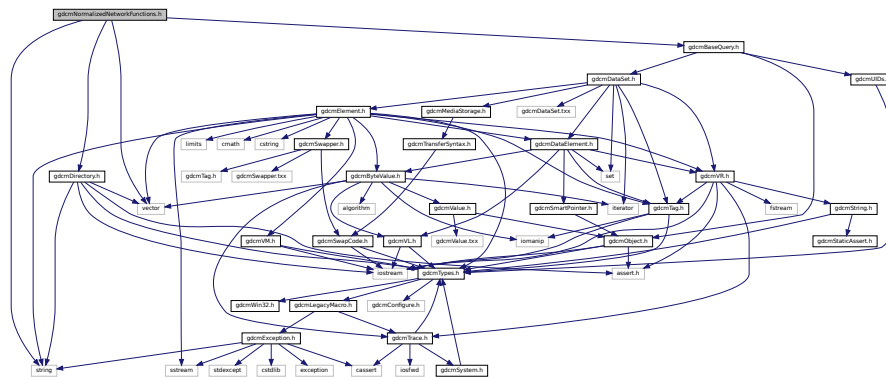
## Namespaces

- `gdc`
- `gdc::network`

## 11.162 gdcmNormalizedNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for `gdcmmNormalizedNetworkFunctions.h`:



## Classes

- class `gdcm::NormalizedNetworkFunctions`  
*Normalized Network Functions.*

## Namespaces

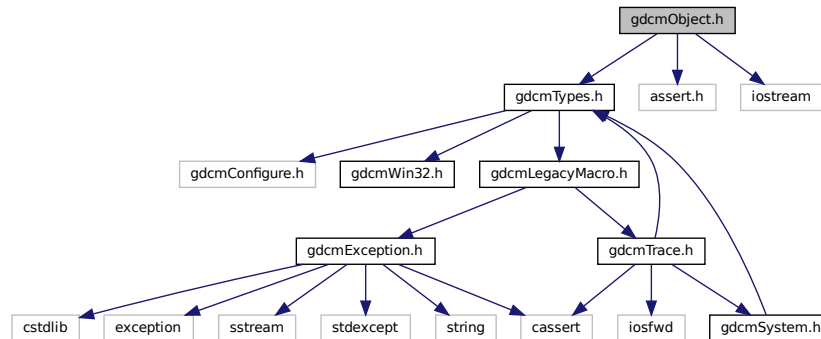
- **gdcm**





```
#include <iostream>
```

Include dependency graph for gdcmObject.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Object`  
*Object.*
- class `gdcm::SmartPointer< ObjectType >`  
*Class for Smart Pointer.*

## Namespaces

- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

## 11.165 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



### Classes

- class [gdcm::OpenSSLCryptoFactory](#)

### Namespaces

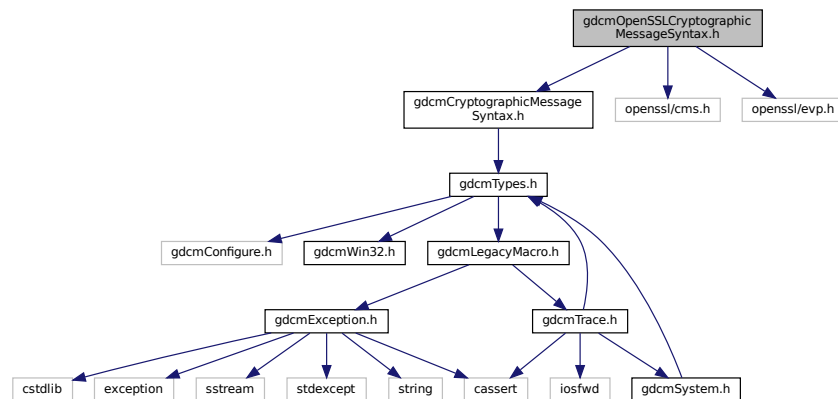
- [gdcm](#)

## 11.166 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

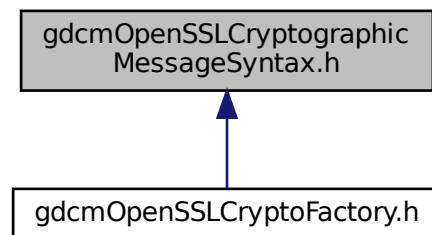
```
#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
```

```
#include <openssl/evp.h>
```

Include dependency graph for gdcmOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::OpenSSLCryptographicMessageSyntax](#)

## Namespaces

- [gdcm](#)

## 11.167 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:
```



### Classes

- class [gdcm::OpenSSLP7CryptoFactory](#)

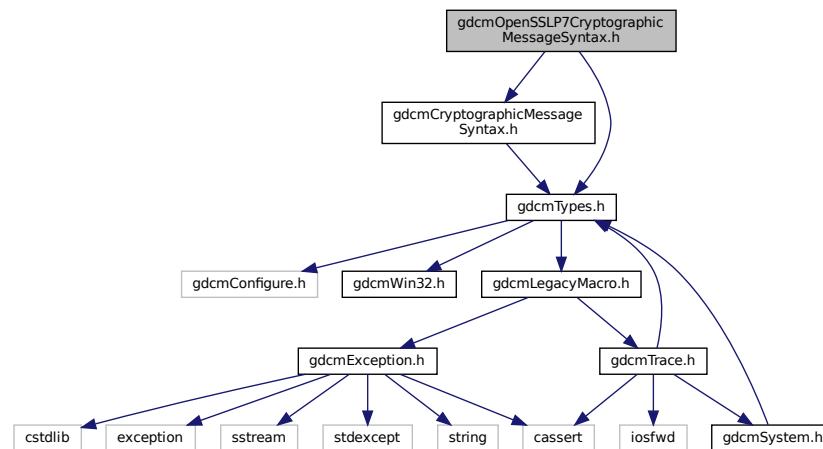
### Namespaces

- [gdcm](#)

## 11.168 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::OpenSSLP7CryptographicMessageSyntax](#)

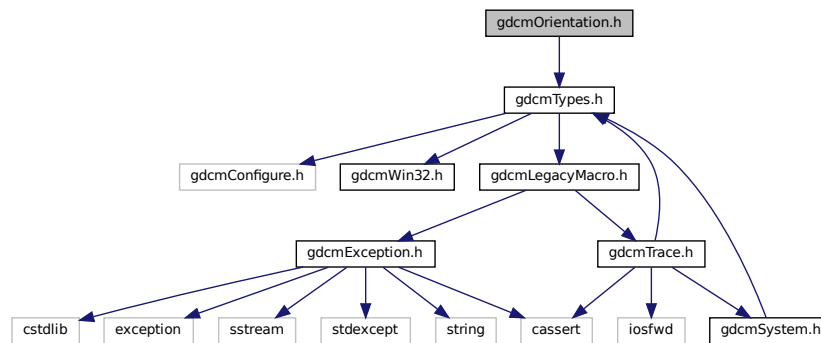
## Namespaces

- [gdcm](#)

## 11.169 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



### Classes

- class [gdcm::Orientation](#)  
class to handle [Orientation](#)

### Namespaces

- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)`

## 11.170 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmObject.h"
```

Include dependency graph for `gdcmOverlay.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::Overlay`  
*Overlay* class.

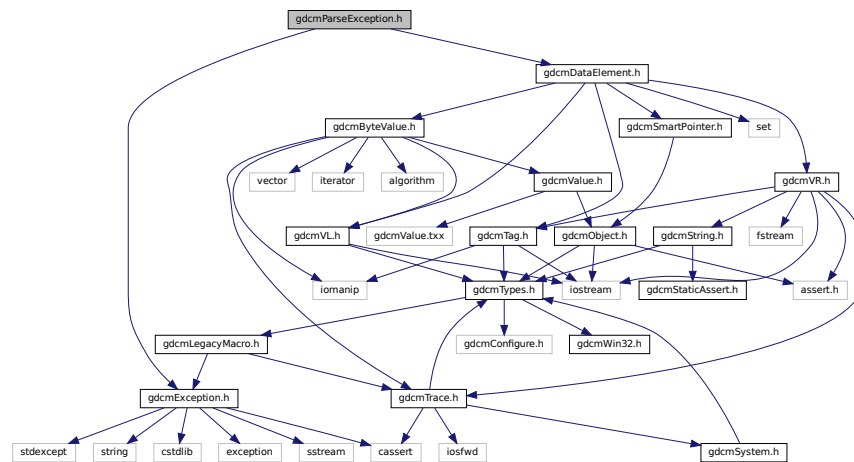
## Namespaces

- `gdcm`

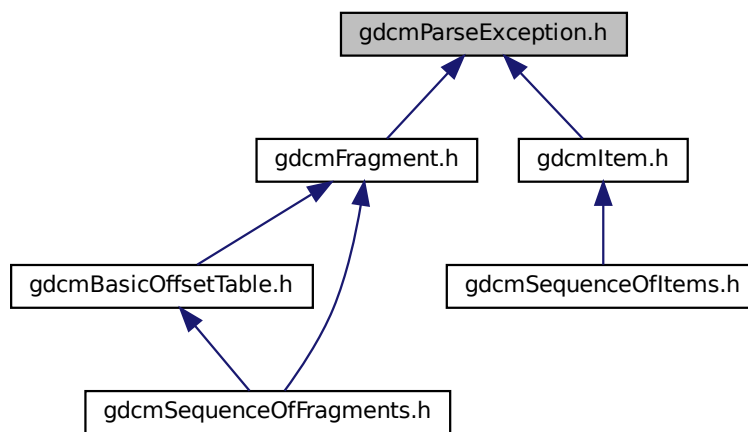
## 11.171 `gdcmParseException.h` File Reference

```
#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

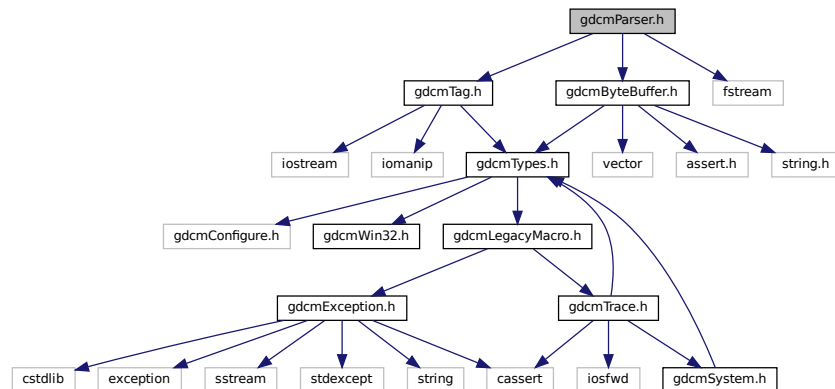
- [gdcm](#)



## 11.172 gdcmParser.h File Reference

```
#include "gdcmTag.h"
#include "gdcmByteBuffer.h"
#include <fstream>
```

Include dependency graph for gdcmParser.h:



### Classes

- class [gdcm::Parser](#)  
*Parser ala XML\_Parser from expat (SAX)*

### Namespaces

- [gdcm](#)

## 11.173 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStudy.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

- [gdcm](#)

## 11.174 gdcmPDataTFPDU.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

- [gdcm](#)
- [gdcm::network](#)

## 11.175 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:





- class `gdc::PDBelement`  
*Class to represent a PDB Element.*

- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const PDBElement &val)`

```
#include "gdcTypes.h"
#include "gdcDataSet.h"
#include "gdcPDBElement.h"
Include dependency graph for gdcPDBHeader.h:
```



## Classes

- class [gdcm::PDBHeader](#)  
*Class for PDBHeader.*

## Namespaces

- [gdcm](#)

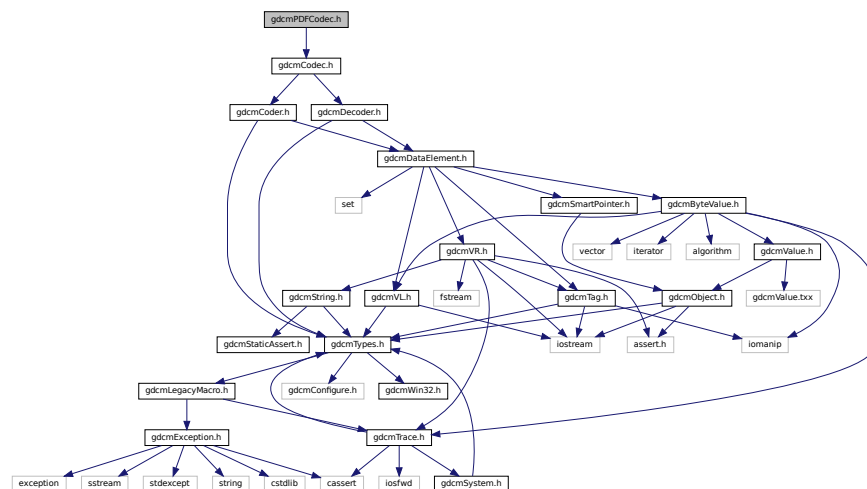
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)`

## 11.177 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmPDFCodec.h:



## Classes

- class [gdcm::PDFCodec](#)  
*PDFCodec class.*

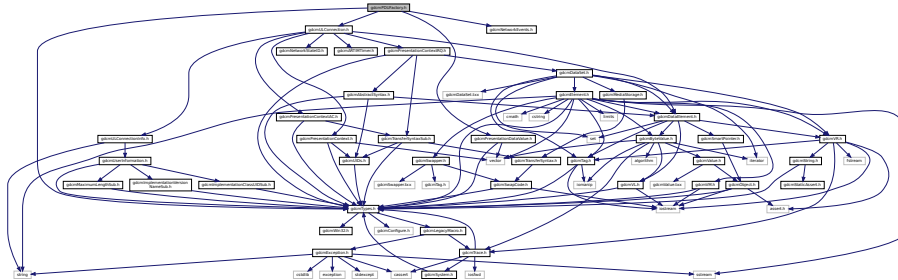
## Namespaces

- [gdcm](#)

## 11.178 gdcmPDUFactory.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

Include dependency graph for gdcmPDUFactory.h:



### Classes

- class [gdcm::network::PDUFactory](#)  
*PDUFactory basically, given an initial byte, construct the.*

### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.179 gdcmPersonName.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

Include dependency graph for gdcmPersonName.h:



## Classes

- class `gdcm::PersonName`  
*PersonName* class.

## Namespaces

- **gdcm**

## 11.180 gdcmPGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcnP GXCodec.h:



## Classes

- class `gdcm::PGXCodec`  
*Class to do PGX.*

## Namespaces

- **gdcm**

## 11.181 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

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)`



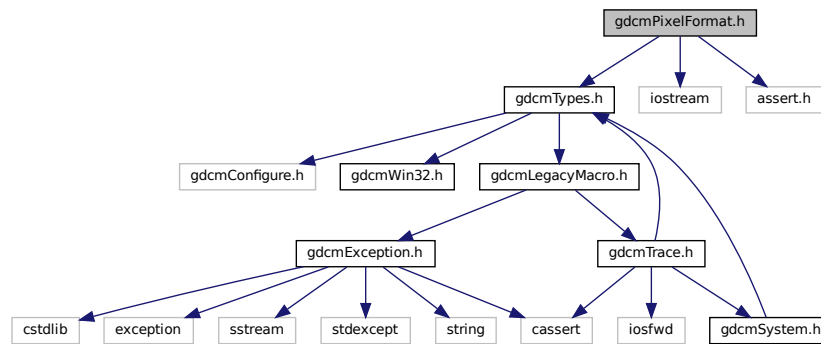
## 11.182 gdcmPixelFormat.h File Reference

```
#include "gdcmTypes.h"
```

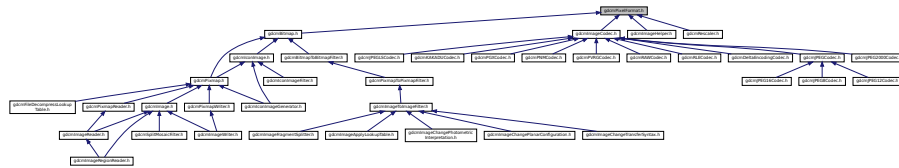
```
#include <iostream>
```

```
#include <assert.h>
```

Include dependency graph for gdcmPixelFormat.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::PixelFormat](#)  
*PixelFormat.*

### Namespaces

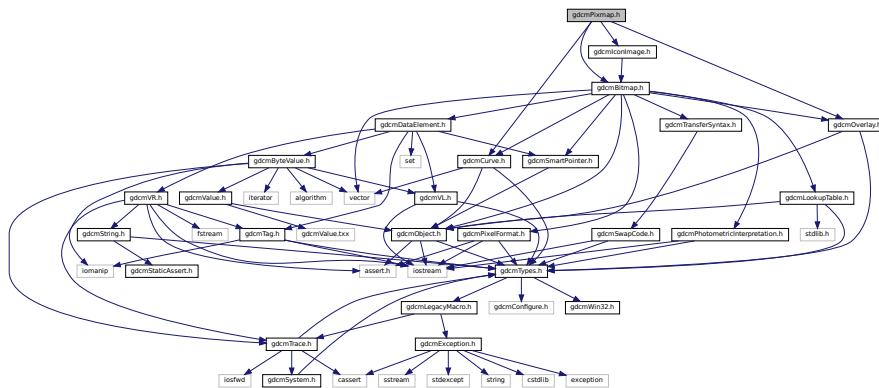
- [gdcm](#)

### Functions

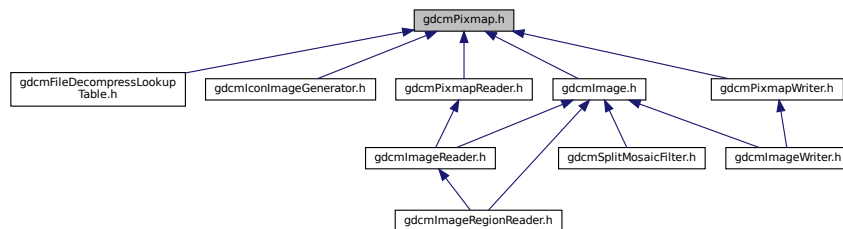
- `std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)`

## 11.183 gdcmPixmap.h File Reference

```
#include "gdcmBitmap.h"
#include "gdcmCurve.h"
#include "gdcmIconImage.h"
#include "gdcmOverlay.h"
Include dependency graph for gdcmPixmap.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::Pixmap](#)  
*Pixmap class.*

### Namespaces

- [gdcm](#)

```
#include "gdcmReader.h"
#include "gdcmPixmap.h"
Include dependency graph for gdcmPixmapReader.h:
```



This graph shows which files directly or indirectly include this file:



- class `gdcm::PixmapReader`  
*PixmapReader*.

- **gdcm**



Include dependency graph for gdcmPixmapWriter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::PixmapWriter`  
*PixmapWriter*.

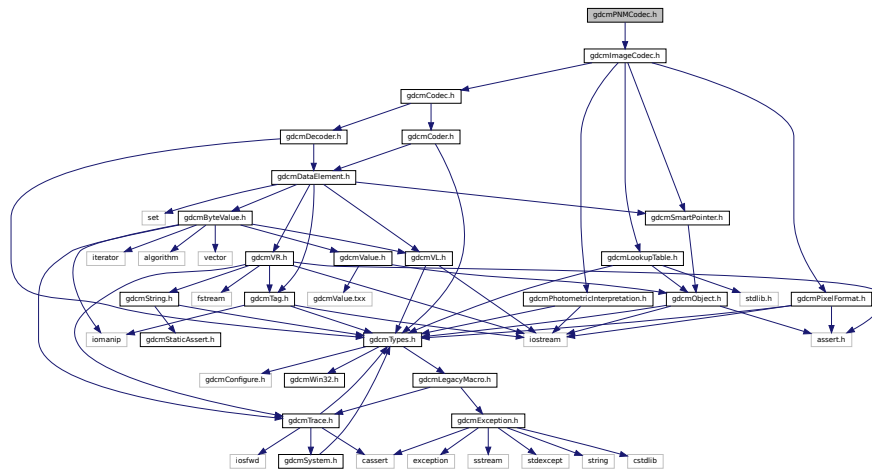
## Namespaces

- **gdcm**

## 11.187 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcnPnmCodec.h:



## Classes

- class `gdcm::PNMCodec`

*Class to do PNM.*

## Namespaces

- gdc

## 11.188 gdcmPreamble.h File Reference

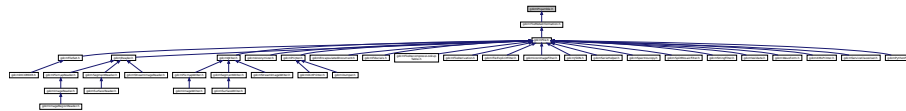
```
#include "gdcmTypes.h"
```

```
#include "gdcml.h"
#include "gdcmlVL.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

- [gdcm](#)

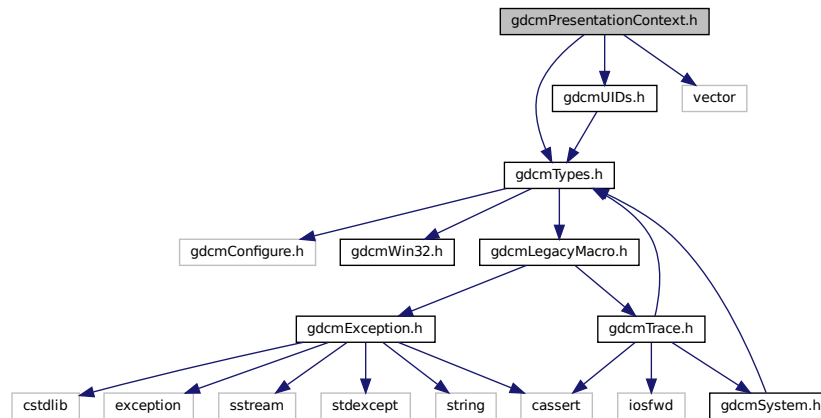
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)`

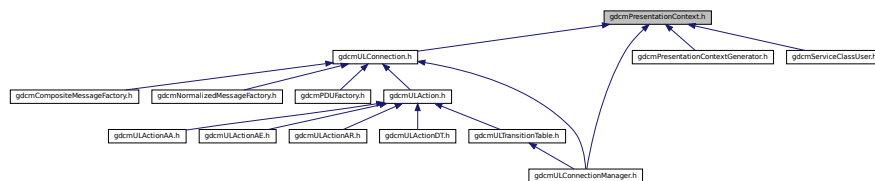
## 11.189 gdcmPresentationContext.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include <vector>
```

Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::PresentationContext`  
*PresentationContext.*

## Namespaces

- `gdcm`

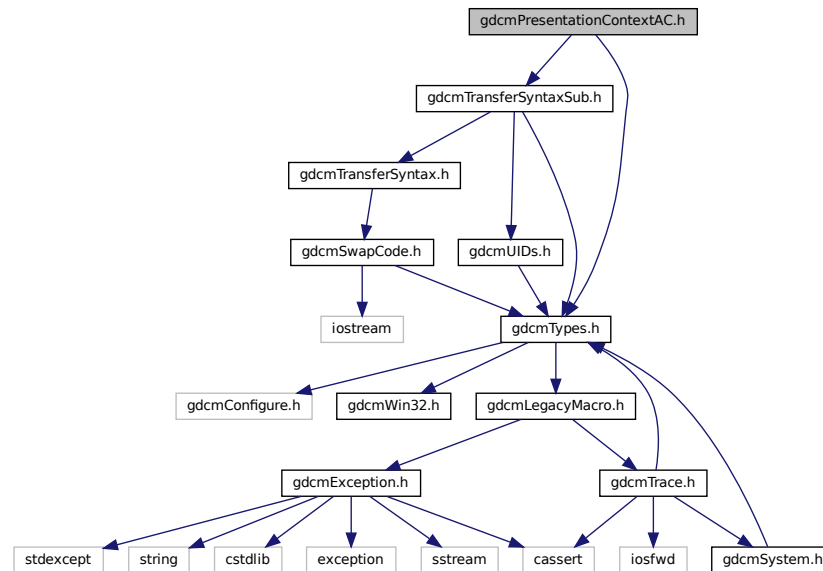


## 11.190 gdcmPresentationContextAC.h File Reference

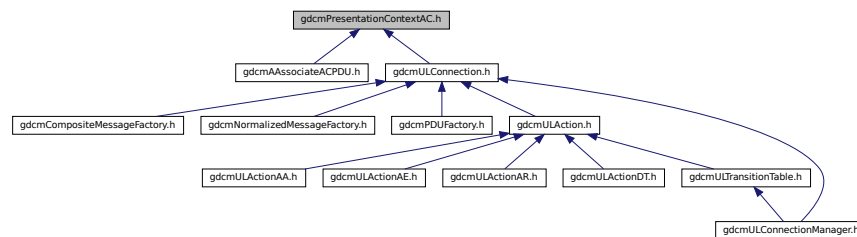
```
#include "gdcmTypes.h"
```

```
#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

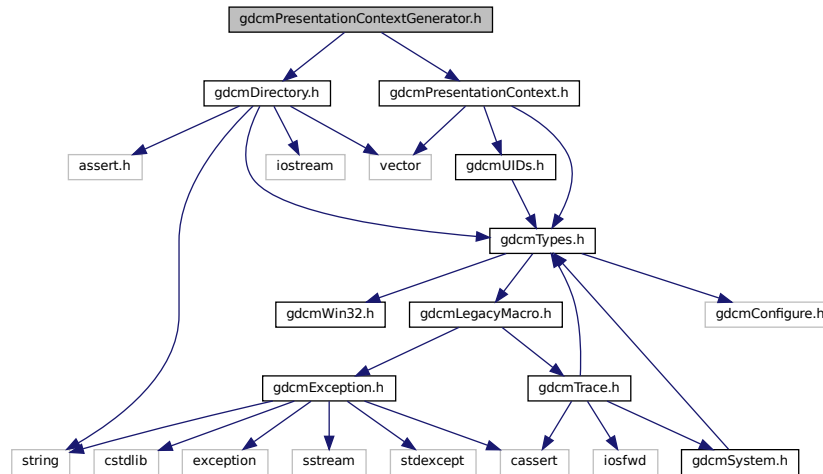
- [gdcm](#)
- [gdcm::network](#)

## 11.191 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
```

```
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



### Classes

- class `gdcm::PresentationContextGenerator`  
*PresentationContextGenerator.*

### Namespaces

- `gdcm`

## 11.192 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmAbstractSyntax.h"
```

```
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for gdcmPresentationContextRQ.h:



This graph shows which files directly or indirectly include this file:



## Classes

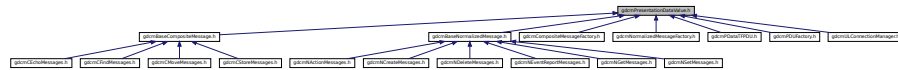
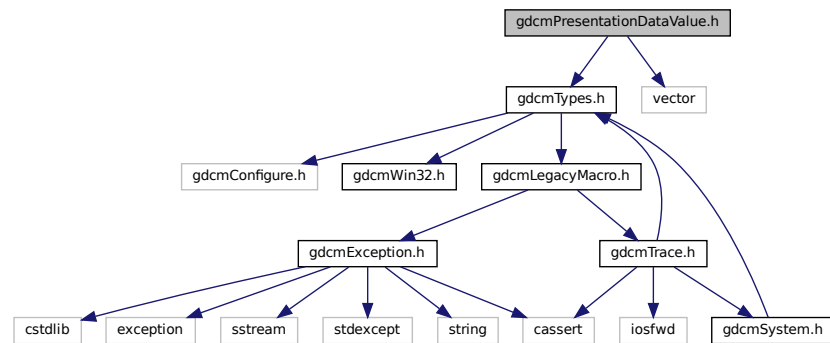
- class `gdcm::network::PresentationContextRQ`  
*PresentationContextRQ*.

## Namespaces

- gdc
- gdc::network

## 11.193 gdcmpresentationdatavalue.h File Reference

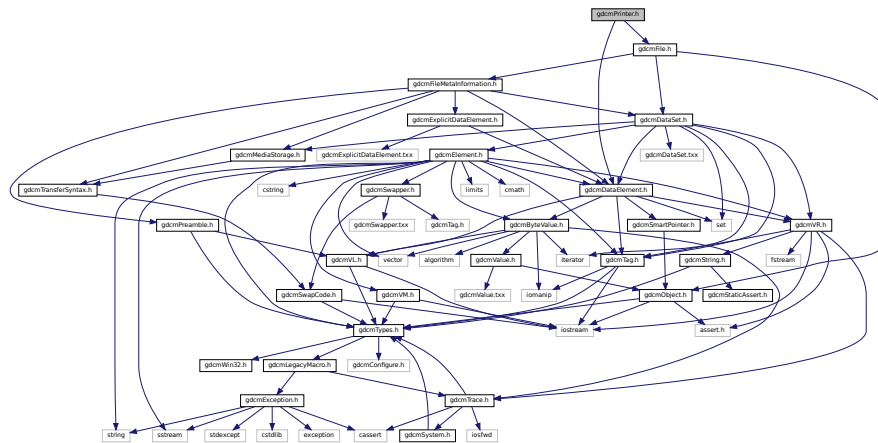
```
#include "gdcmTypes.h"
#include <vector>
```



- class `gdcm::network::PresentationDataValue`  
*PresentationDataValue.*

- `gdcm`
- `gdcm::network`

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```



- class `gdcm::Printer`  
*Printer* class.

- **gdcm**



## Namespaces

- [gdcm](#)

## Functions

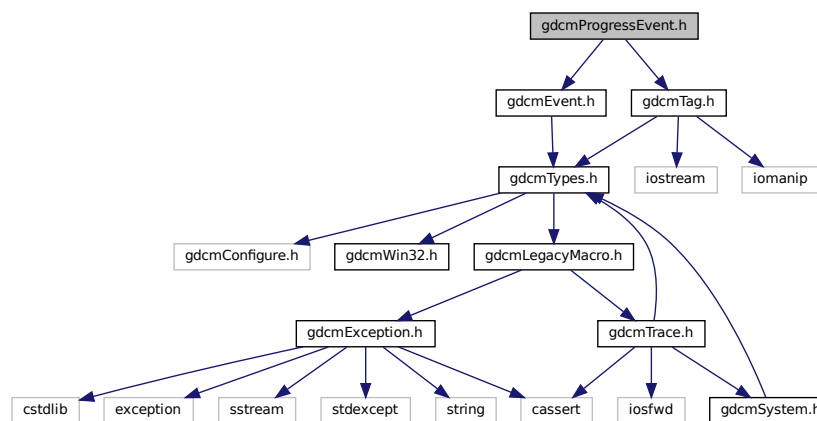
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

## 11.196 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for gdcmProgressEvent.h:



## Classes

- class [gdcm::ProgressEvent](#)  
*ProgressEvent.*

## Namespaces

- [gdcm](#)





## Classes

- class `gdcm::PythonFilter`

`PythonFilter` `PythonFilter` is the class that make `gdcv2.x` looks more like `gdcv1` and transform the binary blob contained in a `DataElement` into a string, typically this is a nice feature to have for wrapped language.

## Namespaces

- **gdcm**

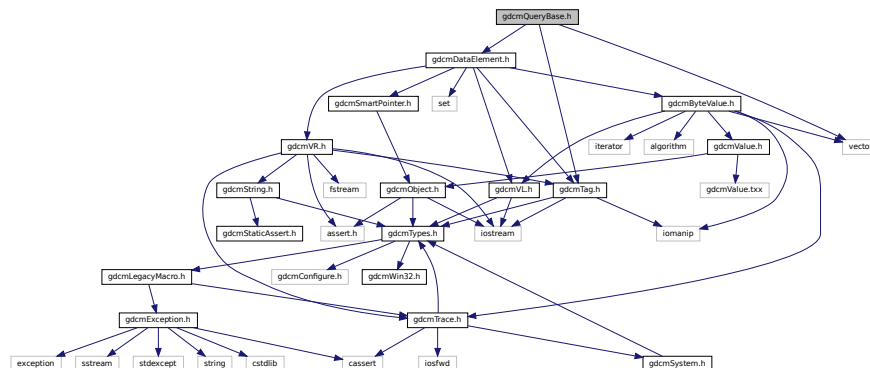
## 11.199 gdcmQueryBase.h File Reference

```
#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

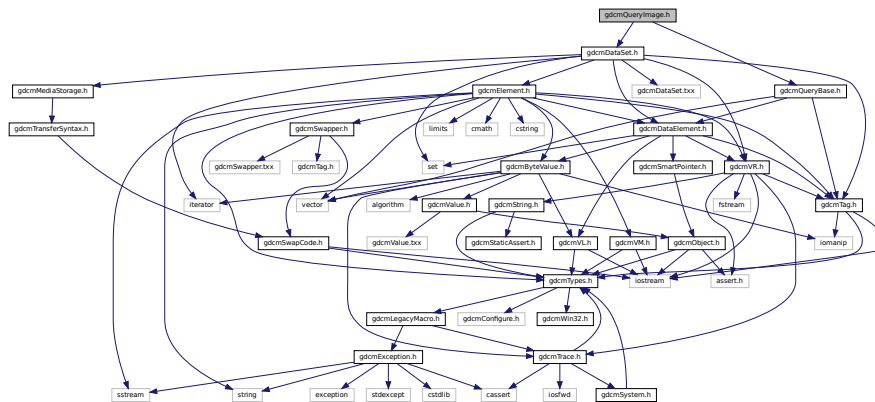
- class `gdcm::QueryBase`

*QueryBase.*



- enum `gdcmm::ECharSet` {  
    `gdcmm::eLatin1` = 0,  
    `gdcmm::eLatin2`,  
    `gdcmm::eLatin3`,  
    `gdcmm::eLatin4`,  
    `gdcmm::eCyrillic`,  
    `gdcmm::eArabic`,  
    `gdcmm::eGreek`,  
    `gdcmm::eHebrew`,  
    `gdcmm::eLatin5`,  
    `gdcmm::eJapanese`,  
    `gdcmm::eThai`,  
    `gdcmm::eJapaneseKanjiMultibyte`,  
    `gdcmm::eJapaneseSupplementaryKanjiMultibyte`,  
    `gdcmm::eKoreanHangulHanjaMultibyte`,  
    `gdcmm::eUTF8`,  
    `gdcmm::eGB18030` }

```
#include "gdcmQueryBase.h"
#include "gdcmDataSet.h"
Include dependency graph for gdcmQueryImage.h:
```



```

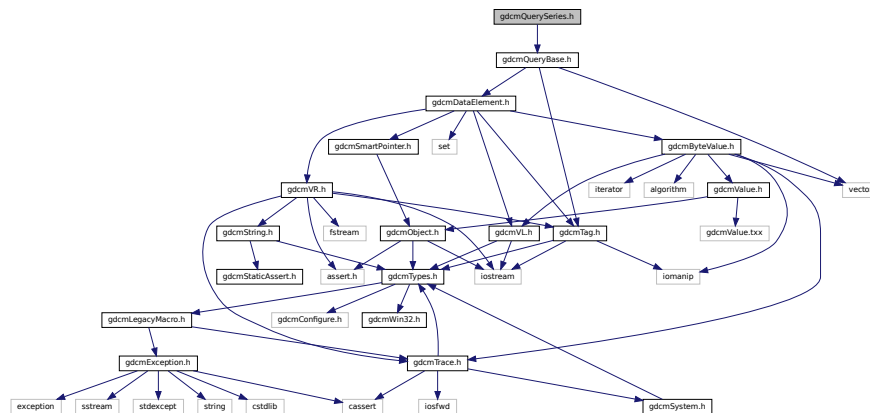
graph TD
    GORM_QueryImage["gorm.QueryImage.h"] --> GORM_BaseRootQuery["gorm.BaseRootQuery.h"]
    GORM_BaseRootQuery --> GORM_BaseComposeMessage["gorm.BaseComposeMessage.h"]
    GORM_BaseRootQuery --> GORM_ComposeNetworkFunctions["gorm.ComposeNetworkFunctions.h"]
    GORM_BaseRootQuery --> GORM_NodeP2PQuery["gorm.NodeP2PQuery.h"]
    GORM_BaseRootQuery --> GORM_NodeStudyQuery["gorm.NodeStudyQuery.h"]
    GORM_BaseRootQuery --> GORM_MoveStudyQuery["gorm.MoveStudyQuery.h"]
    GORM_BaseRootQuery --> GORM_QueryFactory["gorm.QueryFactory.h"]
    GORM_BaseRootQuery --> GORM_WMQuery["gorm.WMQuery.h"]
    GORM_BaseComposeMessage --> GORM_CMMessage1["gorm.CMMessage.h"]
    GORM_BaseComposeMessage --> GORM_CMMessage2["gorm.CMMessage.h"]
    GORM_BaseComposeMessage --> GORM_CMMessage3["gorm.CMMessage.h"]
    GORM_BaseComposeMessage --> GORM_CMMessage4["gorm.CMMessage.h"]
    GORM_NodeP2PQuery --> GORM_MoveQuery["gorm.MoveQuery.h"]
  
```



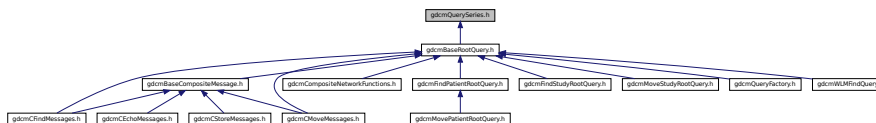
## 11.203 gdcmQuerySeries.h File Reference

```
#include "gdcmQueryBase.h"
```

Include dependency graph for gdcmQuerySeries.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::QuerySeries](#)  
*QuerySeries.*

## Namespaces

- [gdcm](#)









## 11.207 gdcmRegion.h File Reference

```
#include "gdcmTypes.h"
```

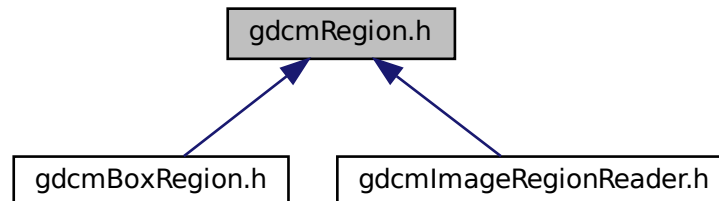
```
#include <vector>
```

```
#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

- [gdcm](#)

## Functions

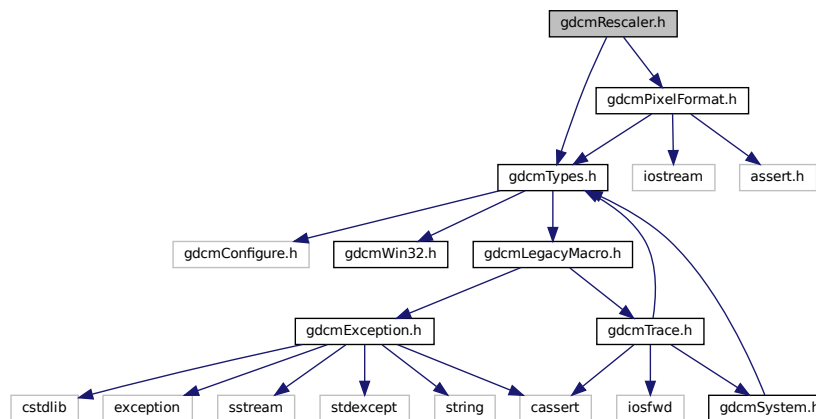
- `std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)`

## 11.208 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for `gdcmRescaler.h`:



## Classes

- class [gdcm::Rescaler](#)

*Rescale class.*

## Namespaces

- [gdcm](#)



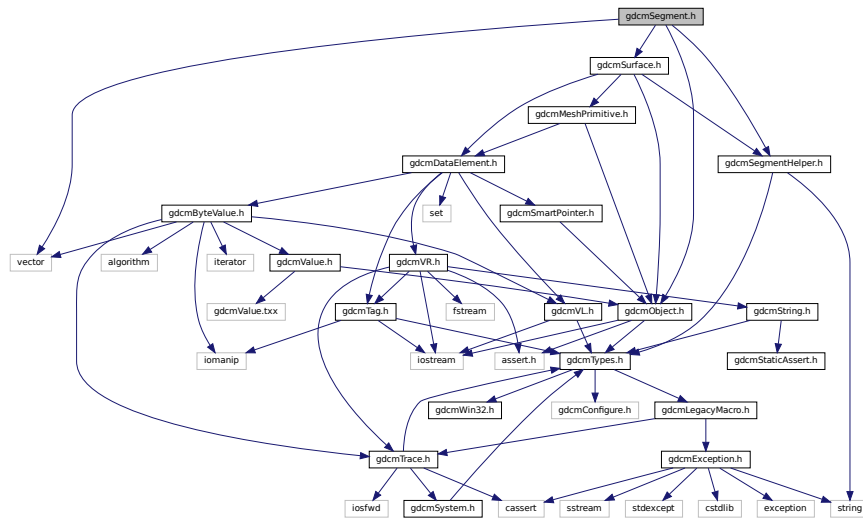


## Functions

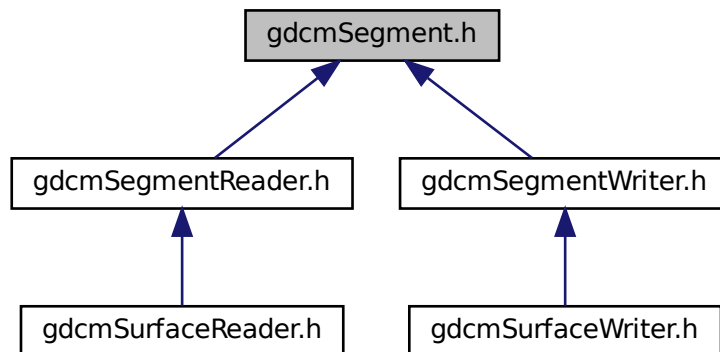
- `std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)`

## 11.212 gdcmSegment.h File Reference

```
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.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.*

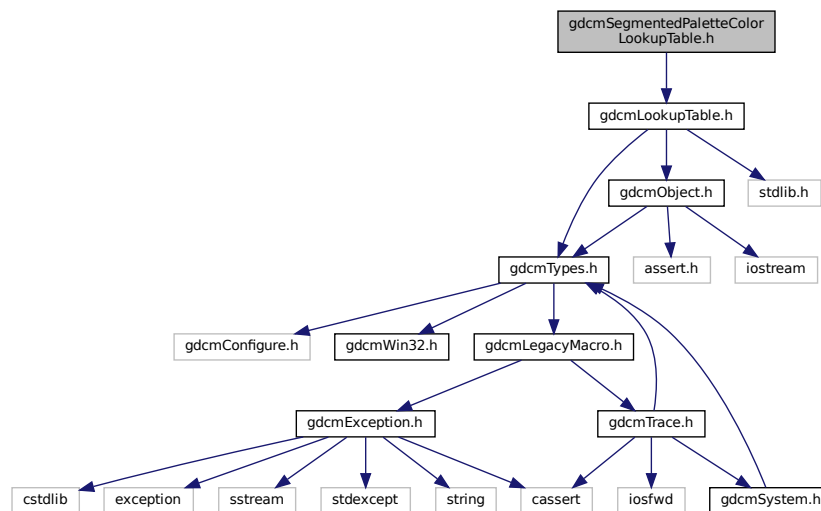
## Namespaces

- [gdcm](#)

## 11.213 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



## Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable class.*

## Namespaces

- [gdcm](#)

## 11.214 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

- [gdcm](#)

## 11.216 gdcmSegmentWriter.h File Reference

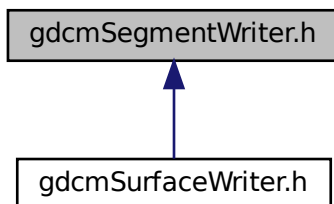
```
#include <gdcmWriter.h>
```

```
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentWriter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::SegmentWriter](#)

*This class defines a segment writer.*

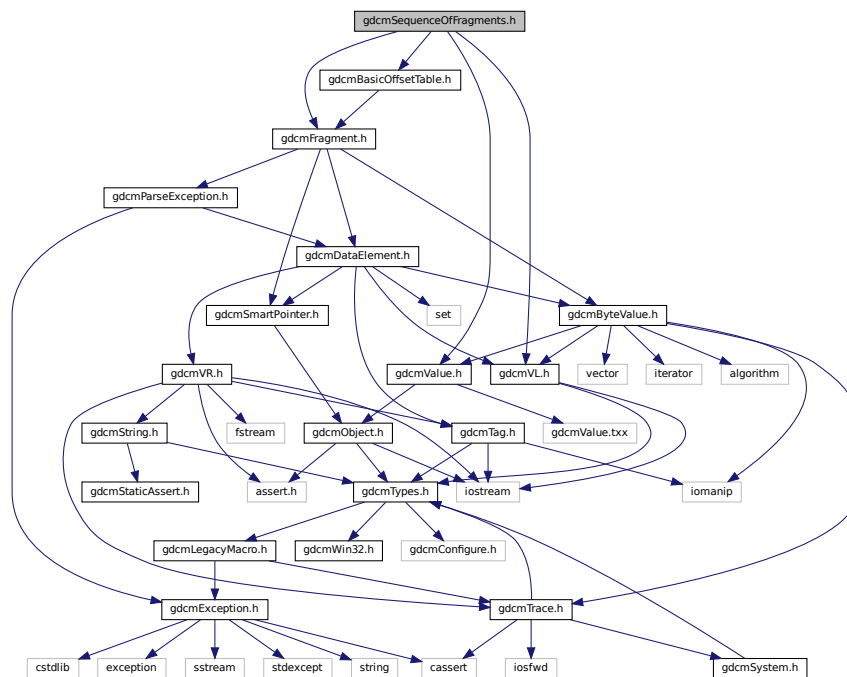
## Namespaces

- [gdcm](#)

## 11.217 gdcmSequenceOfFragments.h File Reference

```
#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
#include "gdcmBasicOffsetTable.h"
```

Include dependency graph for gdcmSequenceOfFragments.h:



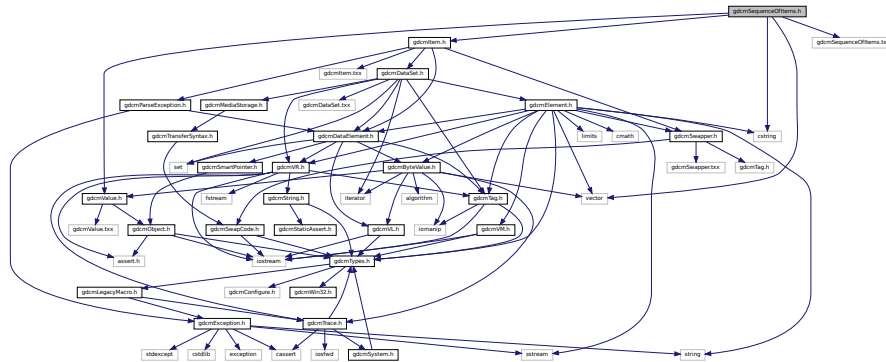
## Classes

- class [gdcm::SequenceOfFragments](#)  
Class to represent a Sequence Of Fragments.

## Namespaces

- [gdcm](#)

```
#include "gdcmlValue.h"
#include "gdcmlItem.h"
#include <vector>
#include <cstring>
#include "gdcmlSequenceOfItems.hxx"
Include dependency graph for gdcmlSequenceOfItems.h:
```



- class `gdcm::SequenceOfItems`  
*Class to represent a Sequence Of Items.*

- **gdcm**

```
#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>
```



## 11.220 gdcmSeries.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSeries.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class `gdcm::Series`  
*Series.*

### Namespaces

- `gdcm`

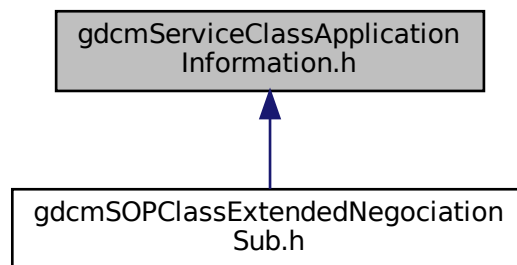
## 11.221 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



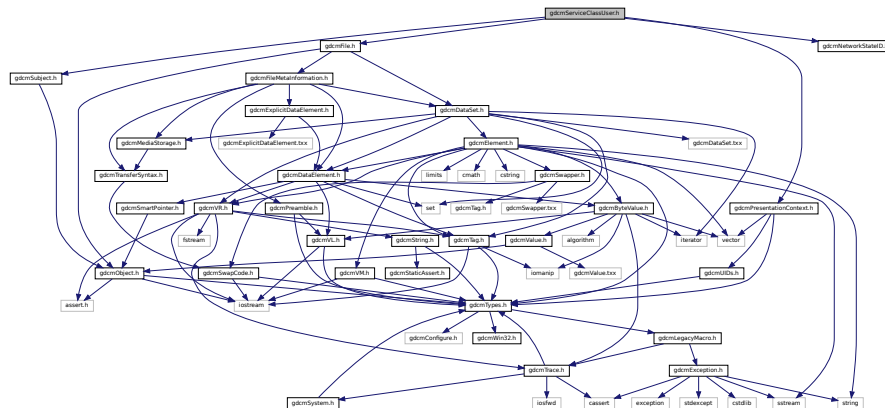
### Classes

- class `gdcm::network::ServiceClassApplicationInformation`

### Namespaces

- `gdcm`
- `gdcm::network`

```
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
#include "gdcmFile.h"
#include "gdcmNetworkStateID.h"
Include dependency graph for gdcmServiceClassUser.h:
```

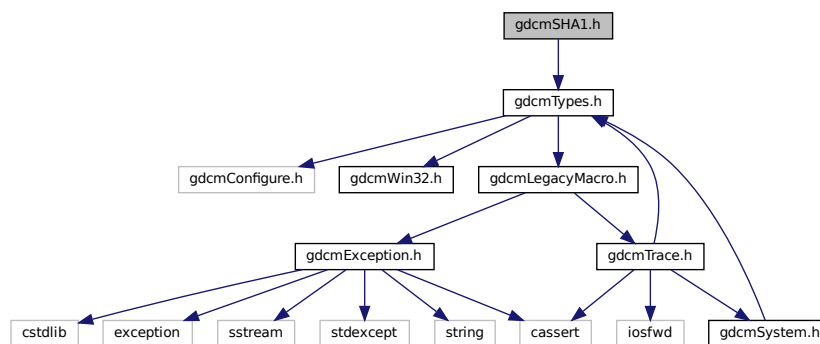


- class `gdcm::ServiceClassUser`  
*ServiceClassUser.*

- `gdcm`
- `gdcm::network`

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmSHA1.h`:



## Classes

- class [gdcm::SHA1](#)

*Class for [SHA1](#).*

## Namespaces

- [gdcm](#)

## 11.224 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 [gdcm::SimpleSubjectWatcher](#)  
*SimpleSubjectWatcher.*

## Namespaces

- [gdcm](#)

## 11.225 gdcmSmartPointer.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for `gdcmSmartPointer.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::SmartPointer< ObjectType >`  
*Class for Smart Pointer.*

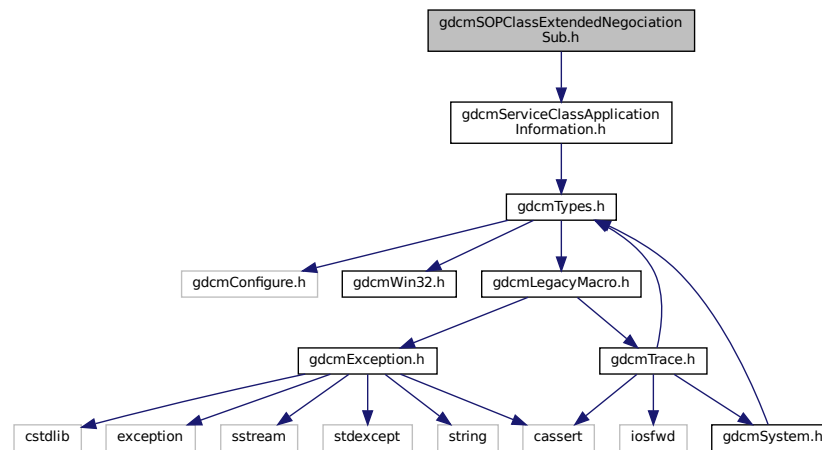
## Namespaces

- `gdcm`

## 11.226 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



## Classes

- class [gdcm::network::SOPClassExtendedNegociationSub](#)  
*SOPClassExtendedNegociationSub.*

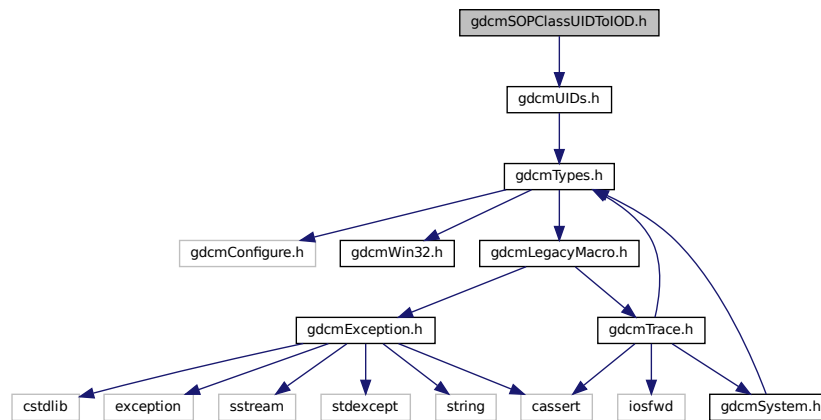
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.227 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

- `gdcm`

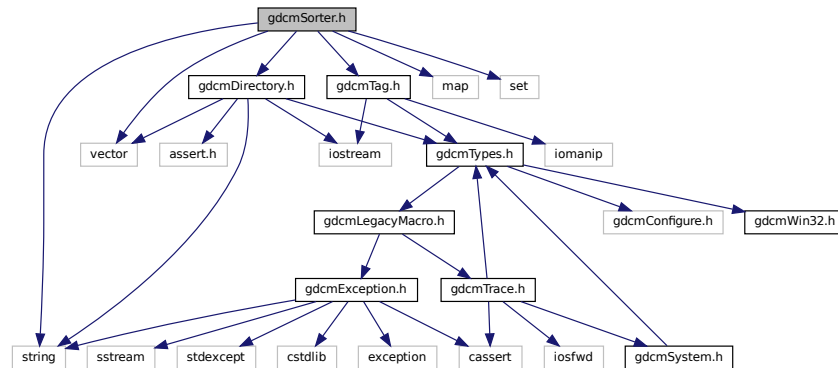
## 11.228 gdcmSorter.h File Reference

```

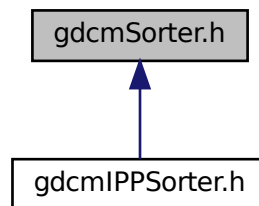
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>
#include <set>

```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Sorter](#)  
*Sorter.*

## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`





## Classes

- struct [gdcmm::static\\_assert\\_test< x >](#)
- struct [gdcmm::STATIC\\_ASSERTION\\_FAILURE< x >](#)
- struct [gdcmm::STATIC\\_ASSERTION\\_FAILURE< true >](#)

## Namespaces

- [gdcmm](#)

## Macros

- [#define GDCM\\_DO\\_JOIN\(X, Y\) GDCM\\_DO\\_JOIN2\(X,Y\)](#)
- [#define GDCM\\_DO\\_JOIN2\(X, Y\) X##Y](#)
- [#define GDCM\\_JOIN\(X, Y\) GDCM\\_DO\\_JOIN\( X, Y \)](#)
- [#define GDCM\\_STATIC\\_ASSERT\(B\)](#)

*The `GDCM_JOIN` + **LINE** is needed to create a uniq identifier.*

### 11.232.1 Macro Definition Documentation

#### 11.232.1.1 GDCM\_DO\_JOIN

```
#define GDCM_DO_JOIN(  
    X,  
    Y ) GDCM\_DO\_JOIN2 (X,Y)
```

#### 11.232.1.2 GDCM\_DO\_JOIN2

```
#define GDCM_DO_JOIN2(  
    X,  
    Y ) X##Y
```

#### 11.232.1.3 GDCM\_JOIN

```
#define GDCM_JOIN(  
    X,  
    Y ) GDCM\_DO\_JOIN ( X, Y )
```



```
#define GDCM_STATIC_ASSERT(  
    B )
```

```
typedef ::gdcmm::static_assert_test<\
    sizeof(::gdcmm::STATIC_ASSERTION_FAILURE< (bool) ( B ) >>)\
    GDCM_JOIN(gdcmm_static_assert_typedef_, __LINE__)
```

## 11.233 gdcMStreamImageReader.h File Reference

Include dependency graph for gdcmStreamImageReader.h:



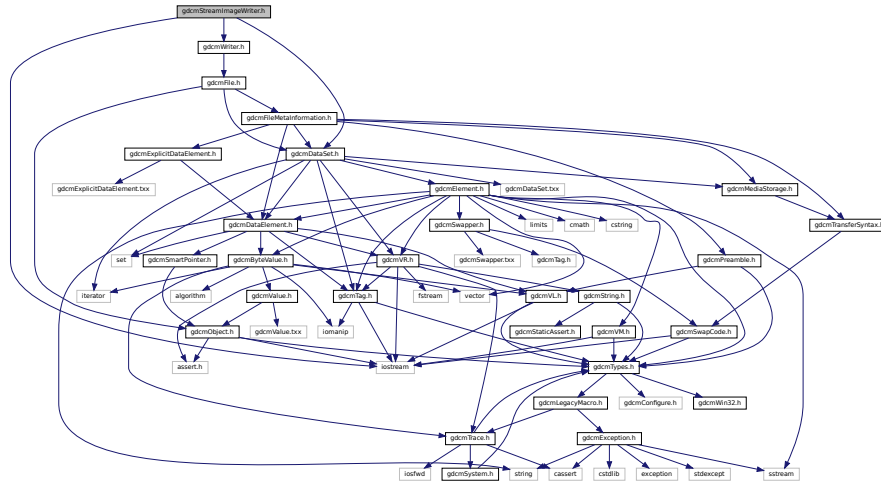
- class `gdcm::StreamImageReader`  
*StreamImageReader*.

- **gdcm**

## 11.234 gdcmStreamImageWriter.h File Reference

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmStreamImageWriter.h:



### Classes

- class [gdcm::StreamImageWriter](#)  
[StreamImageReader](#).

### Namespaces

- [gdcm](#)

## 11.235 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 dependency graph for `gdcmStrictScanner.h`:



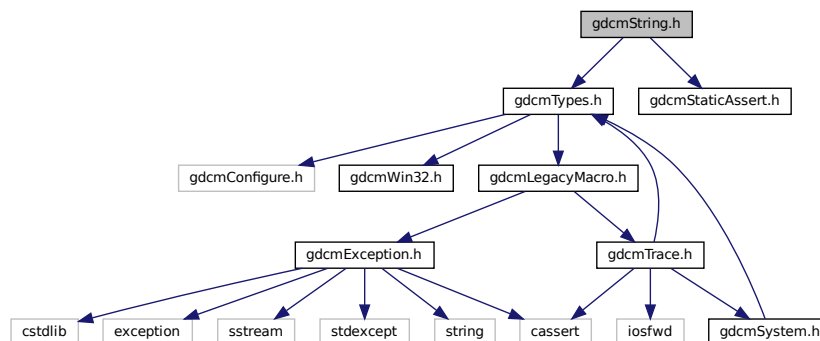
- struct `gdc::StrictScanner::Itstr`
- class `gdc::StrictScanner`  
*StrictScanner.*

- **gdcm**

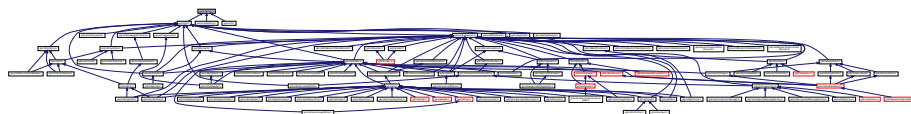
- `std::ostream & gdcmm::operator<< (std::ostream &os, const StrictScanner &s)`

```
#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

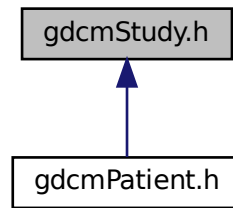
- `gdcm`

## Functions

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



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::Study](#)  
*Study.*

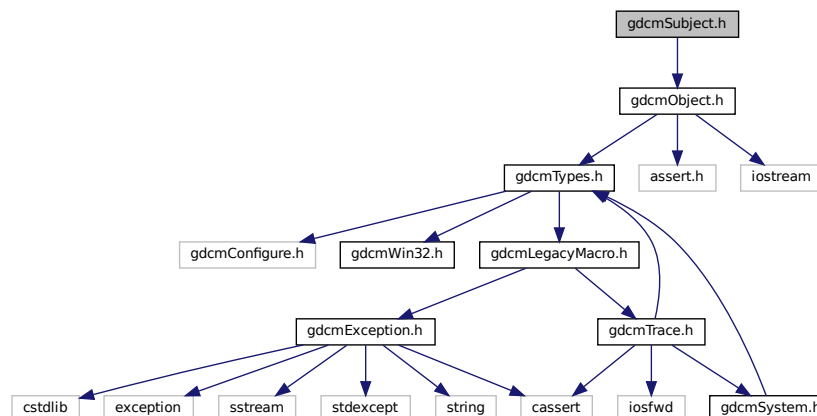
## Namespaces

- [gdc](#)

## 11.239 gdcSubject.h File Reference

```
#include "gdcObject.h"
```

Include dependency graph for gdcSubject.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcms::Surface](#)  
*This class defines a SURFACE IE.*

## Namespaces

- [gdcms](#)

## 11.241 gdcmsurfaceHelper.h File Reference

```
#include "gdcmsTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for `gdcmsurfaceHelper.h`:





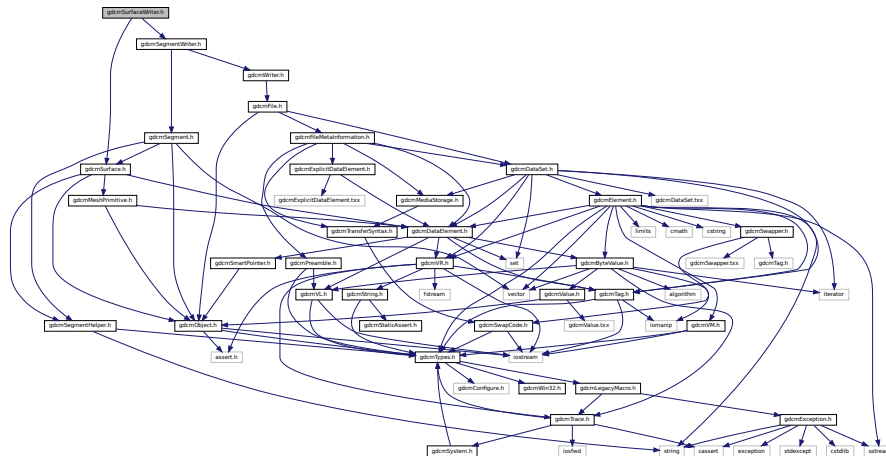


## 11.243 gdcmSurfaceWriter.h File Reference

```
#include <gdcmSegmentWriter.h>
```

```
#include <gdcmSurface.h>
```

Include dependency graph for `gdcmSurfaceWriter.h`:



## Classes

- class `gdcm::SurfaceWriter`

*This class defines a SURFACE IE writer.*

## Namespaces

- **gdcm**

## 11.244 gdcmSwapCode.h File Reference

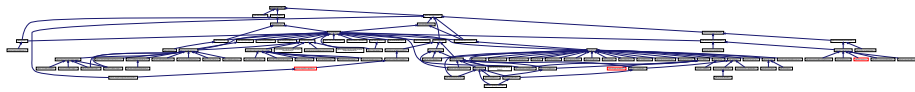
```
#include "gdcmTypes.h"
```

```
#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

- [gdcm](#)

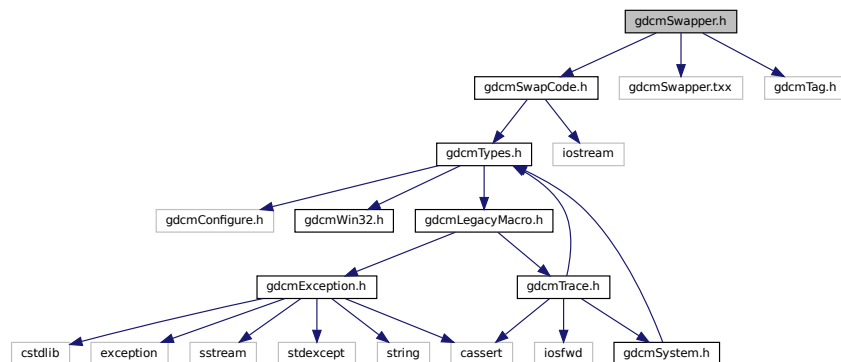
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

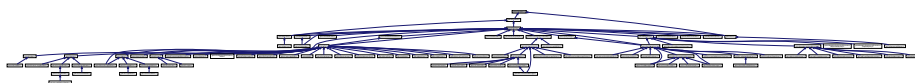
## 11.245 gdcmSwapper.h File Reference

```
#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"
```

Include dependency graph for `gdcmSwapper.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::SwapperDoOp](#)
- class [gdcm::SwapperNoOp](#)

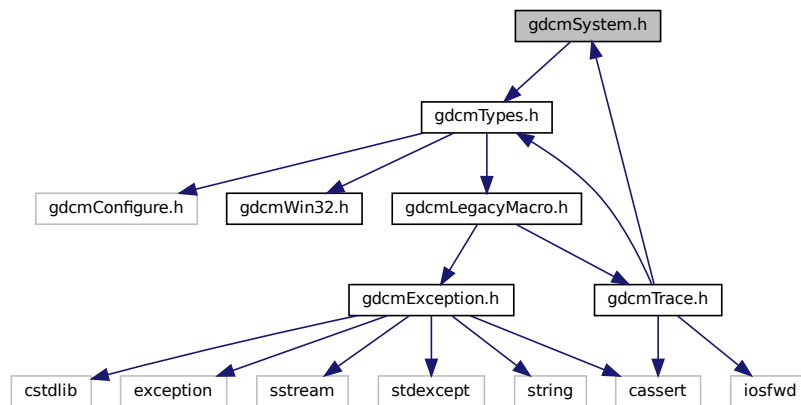
## Namespaces

- [gdcm](#)

## 11.246 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

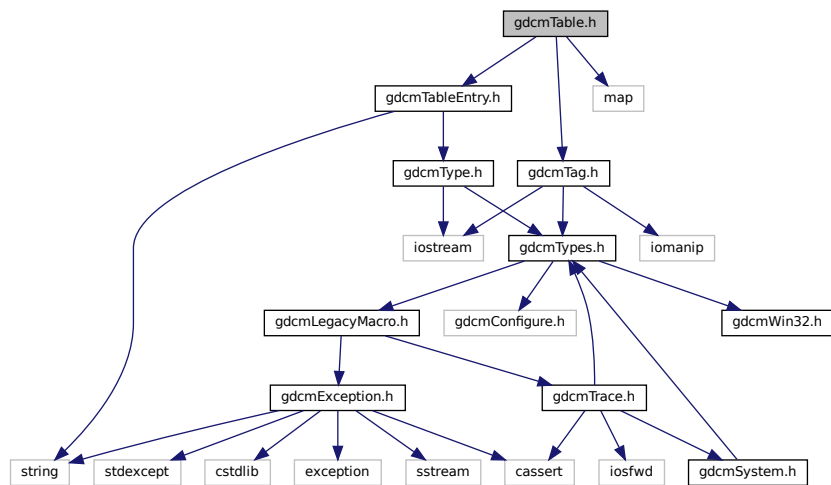
- [gdcm](#)

## 11.247 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
```

```
#include <map>
```

Include dependency graph for `gdcmTable.h`:



## Classes

- class `gdcm::Table`  
*Table.*

## Namespaces

- `gdcm`

## 11.248 gdcmTableEntry.h File Reference

```
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmTableEntry.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::TableEntry`  
*TableEntry*.

## Namespaces

- `gdcm`





## 11.250 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

- [gdcm](#)

### Functions

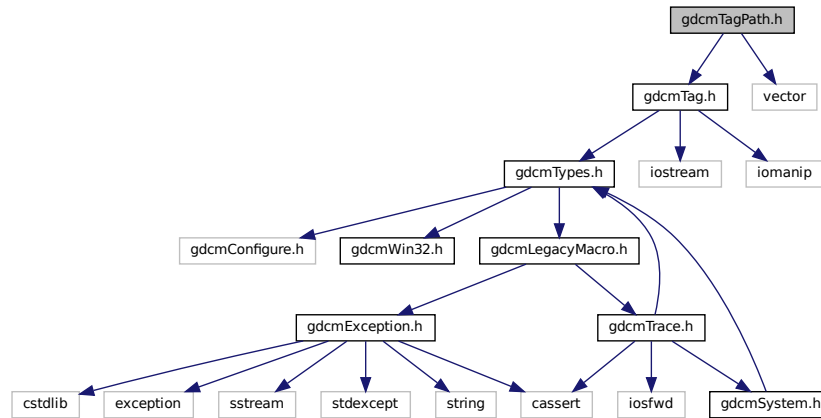
- `std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)`
- `std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)`

## 11.251 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.*

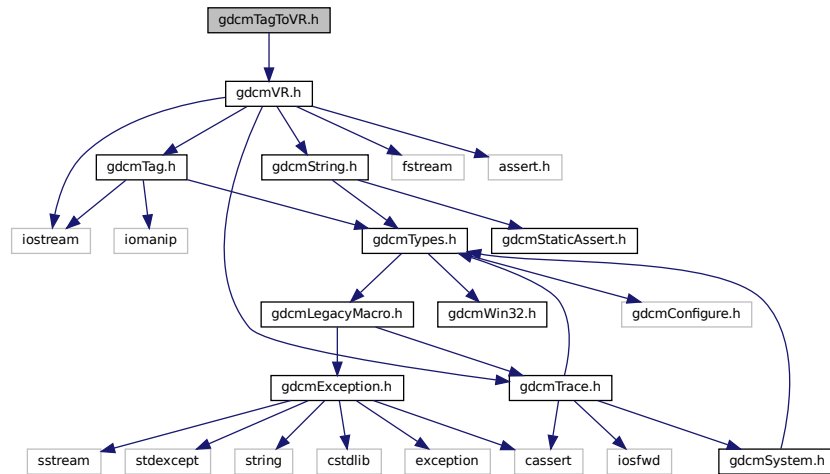
### Namespaces

- [gdcm](#)

## 11.252 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



## Namespaces

- [gdcm](#)

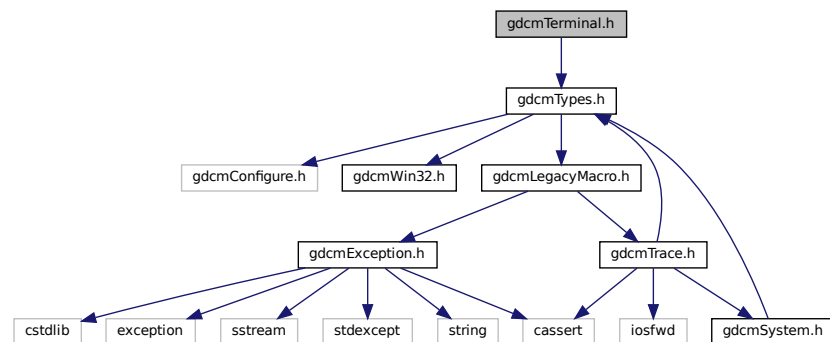
## Functions

- VR::VRType [gdcm::GetVRFromTag](#) (Tag const &tag)

## 11.253 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTerminal.h:



## Namespaces

- [gdc](#)
- [gdc::terminal](#)

*Class for Terminal.*

## Enumerations

- enum [gdc::terminal::Attribute](#) {  
    [gdc::terminal::reset](#) = 0,  
    [gdc::terminal::bright](#) = 1,  
    [gdc::terminal::dim](#) = 2,  
    [gdc::terminal::underline](#) = 3,  
    [gdc::terminal::blink](#) = 5,  
    [gdc::terminal::reverse](#) = 7,  
    [gdc::terminal::hidden](#) = 8 }
- enum [gdc::terminal::Color](#) {  
    [gdc::terminal::black](#) = 0,  
    [gdc::terminal::red](#),  
    [gdc::terminal::green](#),  
    [gdc::terminal::yellow](#),  
    [gdc::terminal::blue](#),  
    [gdc::terminal::magenta](#),  
    [gdc::terminal::cyan](#),  
    [gdc::terminal::white](#) }
- enum [gdc::terminal::Mode](#) {  
    [gdc::terminal::CONSOLE](#) = 0,  
    [gdc::terminal::VT100](#) }

## Functions

- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setattribute](#) (Attribute att)
- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setbgcolor](#) (Color c)
- [GDCM\\_EXPORT](#) std::string [gdc::terminal::setfgcolor](#) (Color c)
- [GDCM\\_EXPORT](#) void [gdc::terminal::setmode](#) (Mode m)

## 11.254 gdcTestDriver.h File Reference

```
#include <clocale>
#include <locale>
```

Include dependency graph for gdcmTestDriver.h:



## 11.255 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



## Classes

- class [gdcm::Testing](#)  
*class for testing*

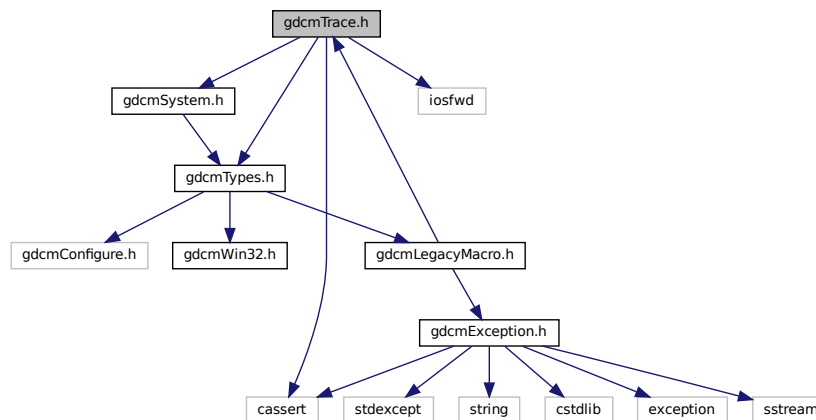
## Namespaces

- [gdcm](#)

## 11.256 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

- `gdcm`

### Macros

- `#define GDCM_FUNCTION` "<unknown>"
- `#define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(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.*

## 11.256.1 Macro Definition Documentation

### 11.256.1.1 GDCM\_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

### 11.256.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(  
    arg ) gdcmAssertMacro(arg)
```

AssertAlways.

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------	---

Referenced by `gdcm::DataElement::GetValue()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::DataSet::Replace()`, `gdcm::DataSet::ReplaceEmpty()`, and `gdcm::VR::Write()`.

### 11.256.1.3 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 );
}

```

Assert.

#### Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------	---

Referenced by `gdcm::PixelFormat::SetSamplesPerPixel()`.

#### 11.256.1.4 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;
    }
}

```

Debug.

#### Parameters

<i>msg</i>	message part
------------	--------------

Referenced by `gdcm::ByteValue::ByteValue()`, `gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory()`, `gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory()`, `gdcm::BasicOffsetTable::Read()`, `gdcm::Item::Read()`, `gdcm::SequenceOfItems::Read()`, `gdcm::VR::Read()`, `gdcm::SequenceOfFragments::ReadPreValue()`, and `gdcm::SequenceOfFragments::ReadValue()`.



## 11.256.1.5 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;
    }
}
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

## Parameters

<i>msg</i>	second message part
------------	---------------------

Referenced by `gdcm::CryptoFactory::CryptoFactory()`, `gdcm::CommandDataSet::Insert()`, `gdcm::FileMetaInformation::Insert()`, `gdcm::DataSet::Insert()`, `gdcm::Item::Read()`, and `gdcm::Fragment::ReadBacktrack()`.

## 11.256.1.6 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;
    }
}
```

Warning.

## Parameters

<i>msg</i>	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()`.

## 11.257 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

- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

## 11.258 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

- [gdcm](#)
- [gdcm::network](#)

## 11.259 gdcmType.h File Reference

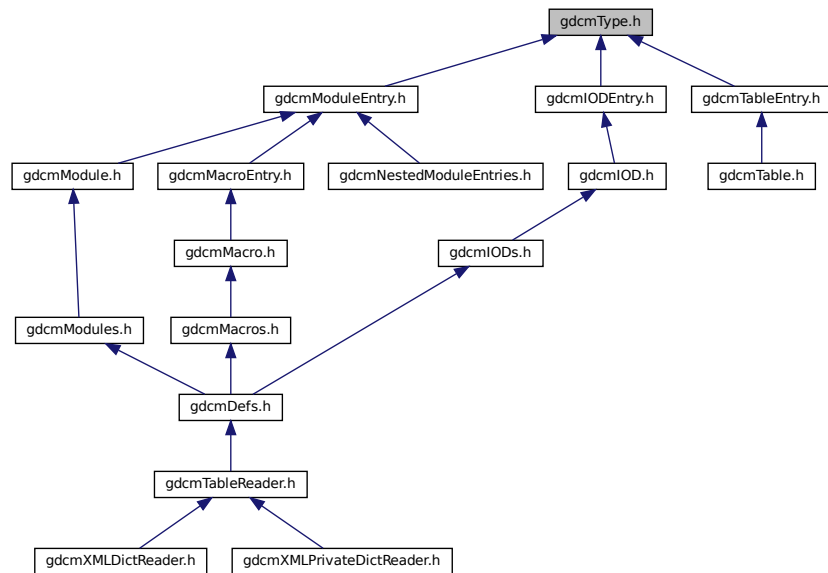
```
#include "gdcmTypes.h"
```

```
#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

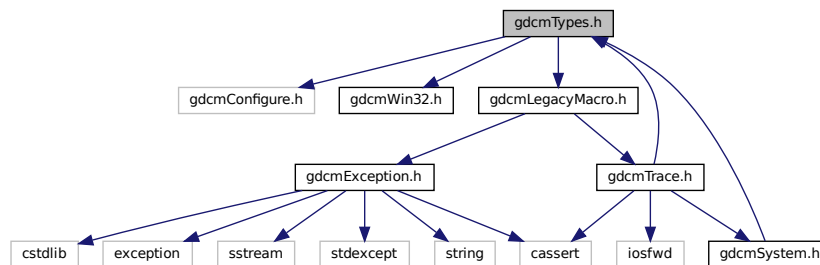
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

## 11.260 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
Include dependency graph for gdcmTypes.h:
```



## 11.261 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
Include dependency graph for gdcmUIDGenerator.h:
```



## Classes

- class [gdcm::UIDGenerator](#)  
*Class for generating unique UID.*

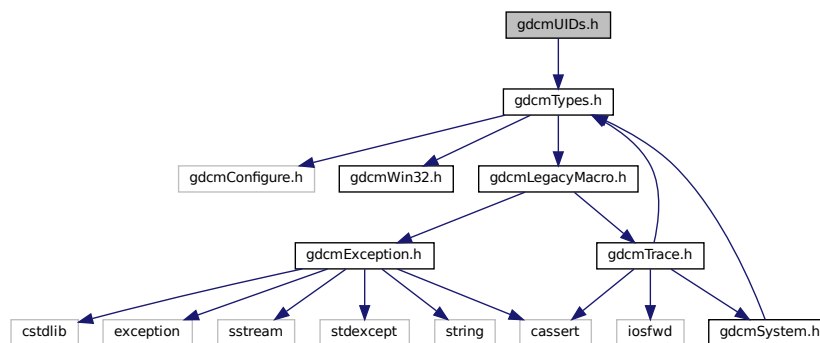
## Namespaces

- [gdcm](#)

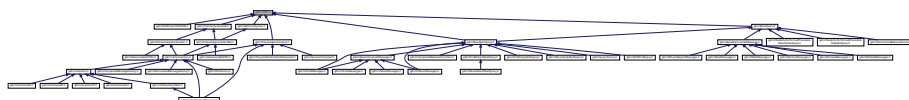
## 11.262 gdcmUIDs.h File Reference

```
#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

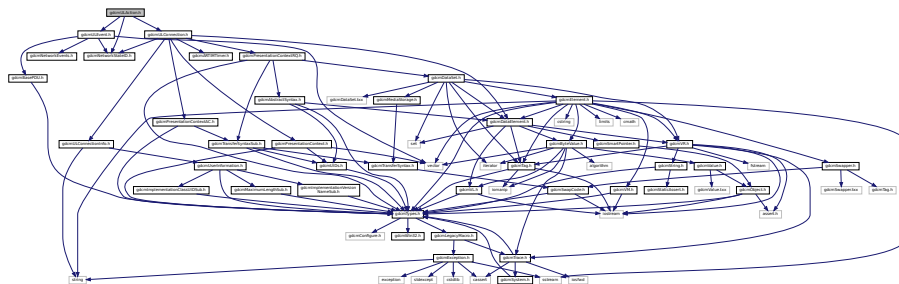
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

## 11.263 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.*

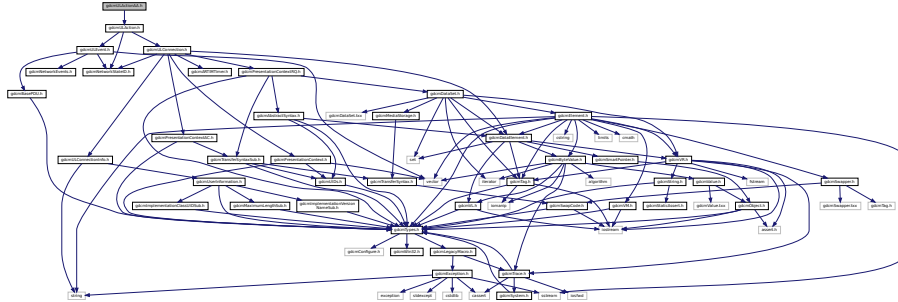
## Namespaces

- `gdcm`
- `gdcm::network`

## 11.264 gdcmULActionAA.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionAA.h:



### Classes

- class [gdcm::network::ULActionAA1](#)
- class [gdcm::network::ULActionAA2](#)
- class [gdcm::network::ULActionAA3](#)
- class [gdcm::network::ULActionAA4](#)
- class [gdcm::network::ULActionAA5](#)
- class [gdcm::network::ULActionAA6](#)
- class [gdcm::network::ULActionAA7](#)
- class [gdcm::network::ULActionAA8](#)

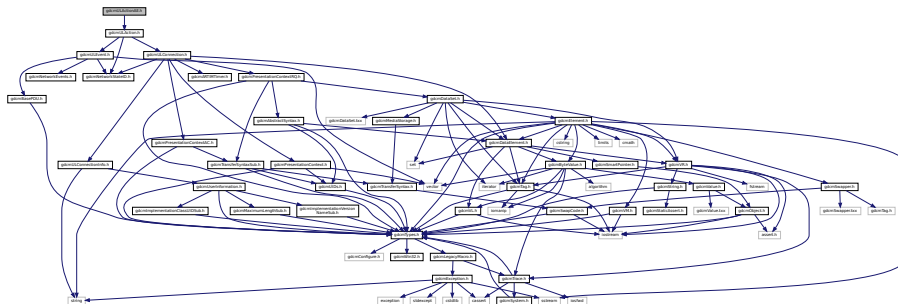
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.265 gdcmULActionAE.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionAE.h:





## Classes

- class [gdcm::network::ULActionAE1](#)
- class [gdcm::network::ULActionAE2](#)
- class [gdcm::network::ULActionAE3](#)
- class [gdcm::network::ULActionAE4](#)
- class [gdcm::network::ULActionAE5](#)
- class [gdcm::network::ULActionAE6](#)
- class [gdcm::network::ULActionAE7](#)
- class [gdcm::network::ULActionAE8](#)

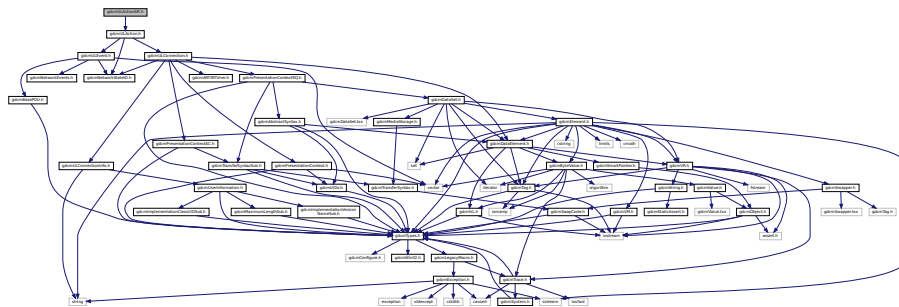
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.266 gdcmULActionAR.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionAR.h:



## Classes

- class [gdcm::network::ULActionAR1](#)
- class [gdcm::network::ULActionAR10](#)
- class [gdcm::network::ULActionAR2](#)
- class [gdcm::network::ULActionAR3](#)
- class [gdcm::network::ULActionAR4](#)
- class [gdcm::network::ULActionAR5](#)
- class [gdcm::network::ULActionAR6](#)
- class [gdcm::network::ULActionAR7](#)
- class [gdcm::network::ULActionAR8](#)
- class [gdcm::network::ULActionAR9](#)

## Namespaces

- [gdc](#)
- [gdc::network](#)

## 11.267 gdcULActionDT.h File Reference

```
#include "gdcULAction.h"
```

Include dependency graph for gdcULActionDT.h:



## Classes

- class [gdc::network::ULActionDT1](#)
- class [gdc::network::ULActionDT2](#)

## Namespaces

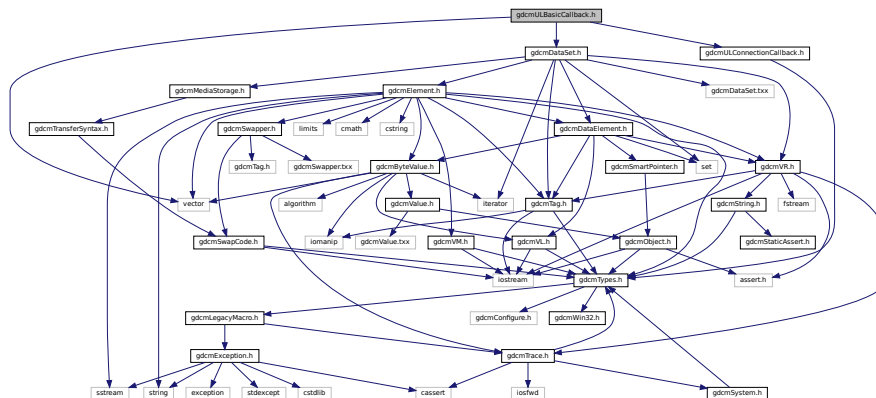
- [gdc](#)
- [gdc::network](#)

## 11.268 gdcULBasicCallback.h File Reference

```
#include "gdcULConnectionCallback.h"
```

```
#include "gdcDataSet.h"
```

Include dependency graph for gdcmlBasicCallback.h:



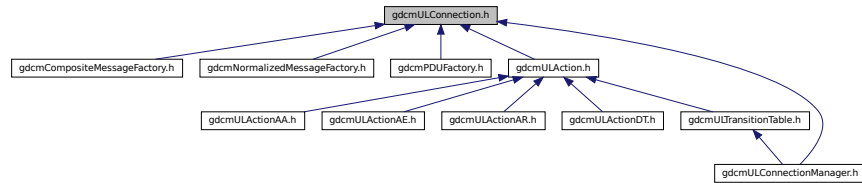
- class `gdcm::network::ULBasicCallback`  
*ULBasicCallback.*

- `gdc`
- `gdc::network`

```
#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 gdcmULConnection.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::ULConnection](#)  
*ULConnection.*

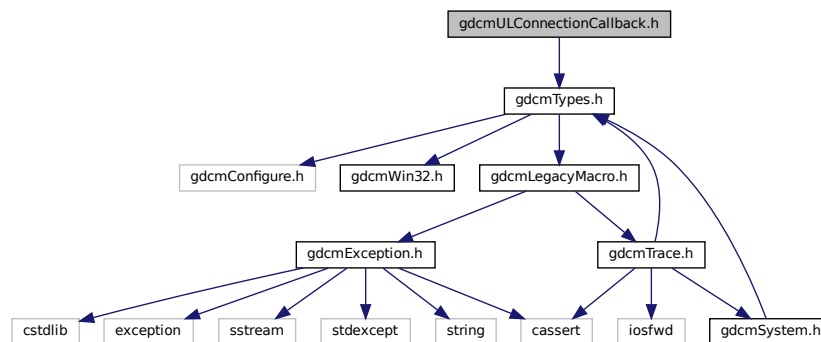
## Namespaces

- [gdcml](#)
- [gdcml::network](#)

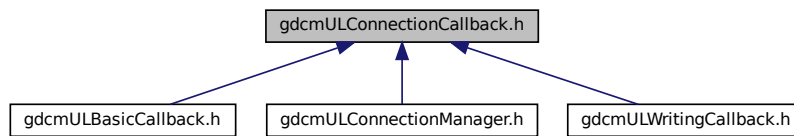
## 11.270 gdcmlULConnectionCallback.h File Reference

```
#include "gdcmlTypes.h"
```

Include dependency graph for gdcmlULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



## Classes

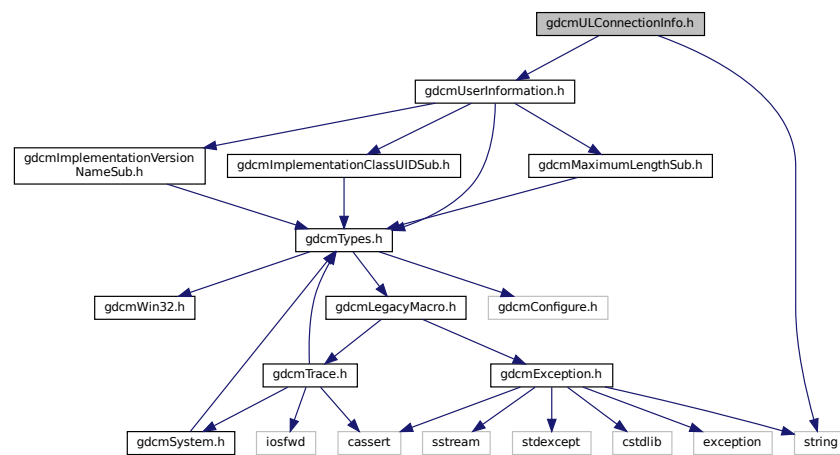
- class `gdcm::network::ULConnectionCallback`

## Namespaces

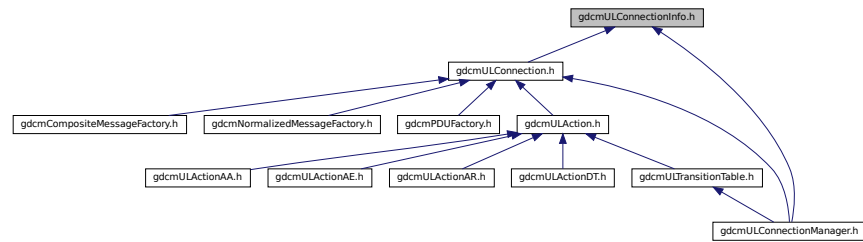
- `gdcm`
- `gdcm::network`

## 11.271 gdcmULConnectionInfo.h File Reference

```
#include "gdcmUserInformation.h"  
#include <string>  
Include dependency graph for gdcmULConnectionInfo.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::ULConnectionInfo](#)  
*ULConnectionInfo*.

## Namespaces

- [gdcml](#)
- [gdcml::network](#)

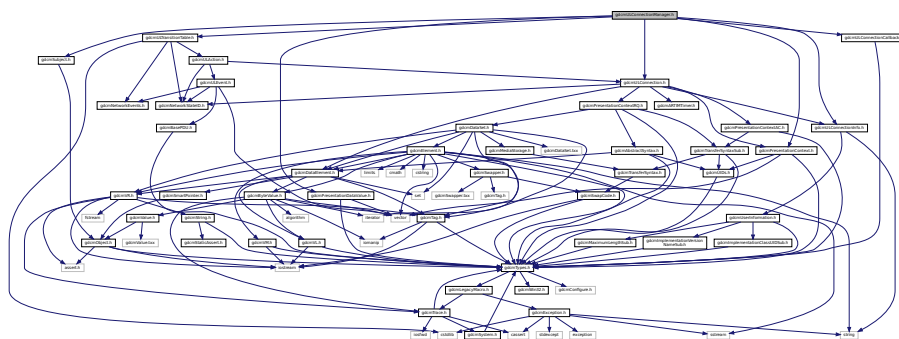
## 11.272 gdcmlULConnectionManager.h File Reference

```

#include "gdcmlULTransitionTable.h"
#include "gdcmlULConnection.h"
#include "gdcmlULConnectionInfo.h"
#include "gdcmlPresentationDataValue.h"
#include "gdcmlULConnectionCallback.h"
#include "gdcmlSubject.h"
#include "gdcmlPresentationContext.h"

```

Include dependency graph for gdcmlULConnectionManager.h:



## Classes

- class [gdcm::network::ULConnectionManager](#)  
*ULConnectionManager.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.273 gdcmULEvent.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmULEvent.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::ULEvent`  
*ULEvent.*

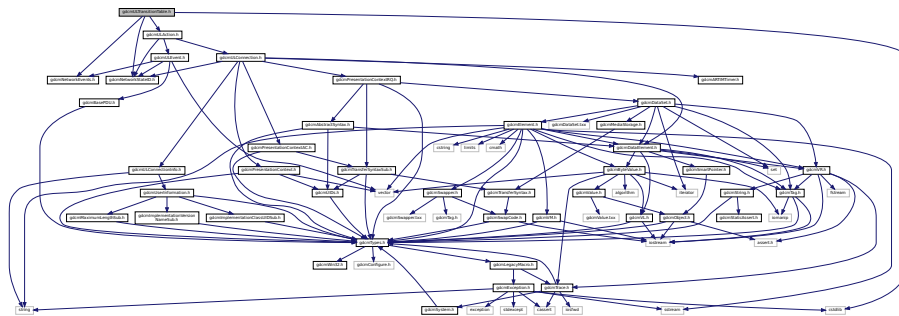
## Namespaces

- `gdcm`
- `gdcm::network`

## 11.274 gdcmULTransitionTable.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULAction.h"
#include <cstdlib>
```

Include dependency graph for `gdcmULTransitionTable.h`:



This graph shows which files directly or indirectly include this file:





## Classes

- class [gdcm::network::TableRow](#)
- struct [gdcm::network::Transition](#)
- class [gdcm::network::ULTransitionTable](#)

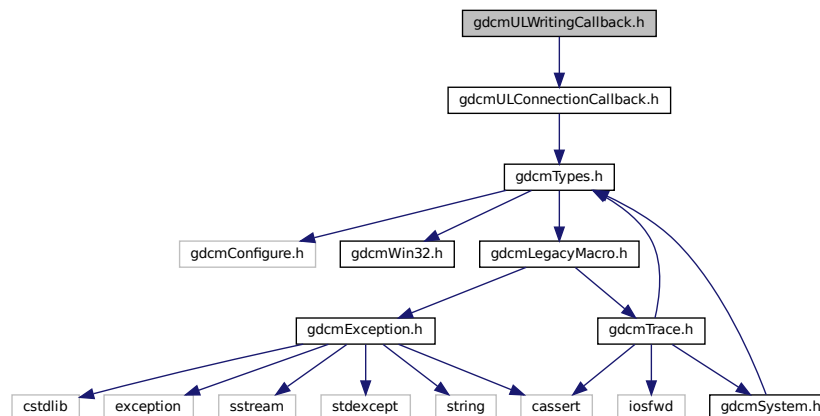
*[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.275 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
Include dependency graph for gdcmULWritingCallback.h:
```



## Classes

- class [gdcm::network::ULWritingCallback](#)

## Namespaces

- [gdcm](#)
- [gdcm::network](#)



## Namespaces

- [gdcm](#)

## 11.278 gdcmUnpacker12Bits.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUnpacker12Bits.h:



## Classes

- class [gdcm::Unpacker12Bits](#)  
Pack/Unpack 12 bits pixel into 16bits.

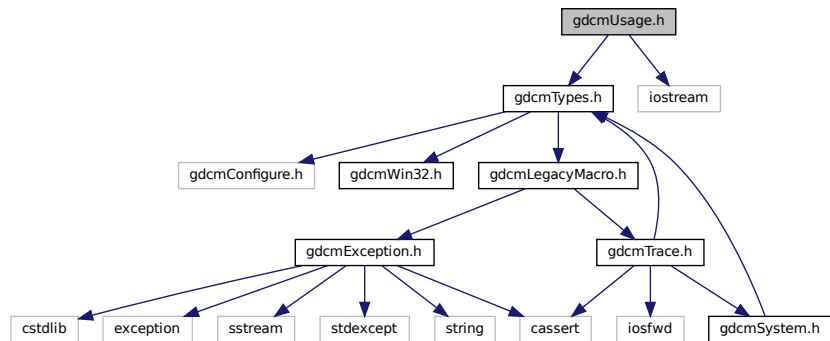
## Namespaces

- [gdcm](#)

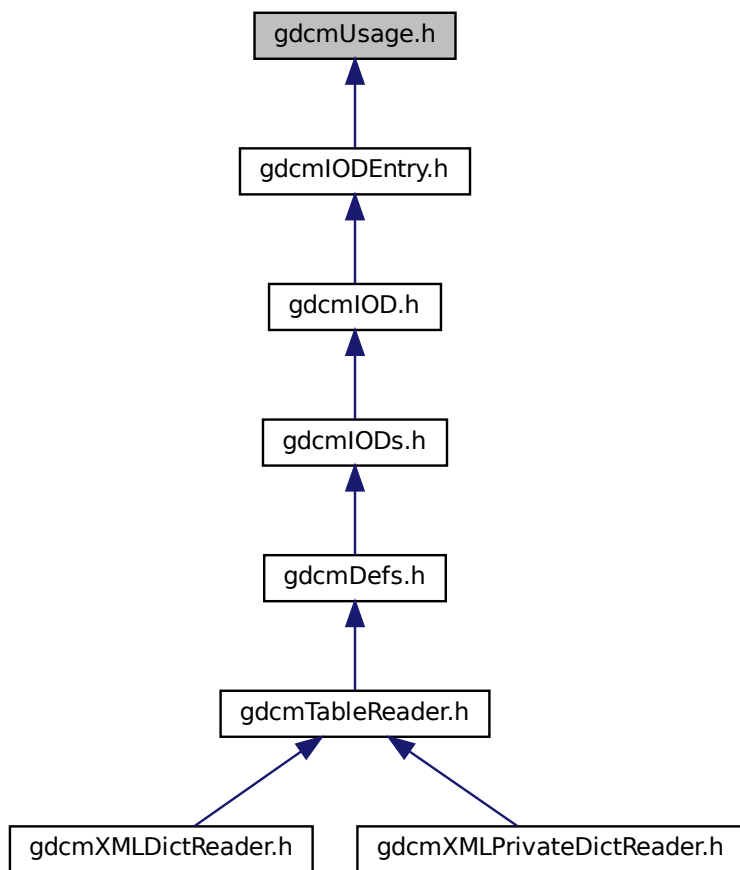
## 11.279 gdcmUsage.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for `gdcmUsage.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Usage](#)  
*Usage.*

## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)`

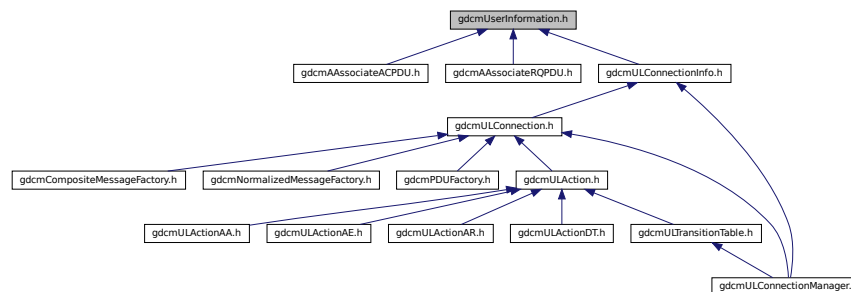
## 11.280 gdcmUserInformation.h File Reference

```
#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 [gdcm::network::UserInfo](#)  
*UserInfo.*

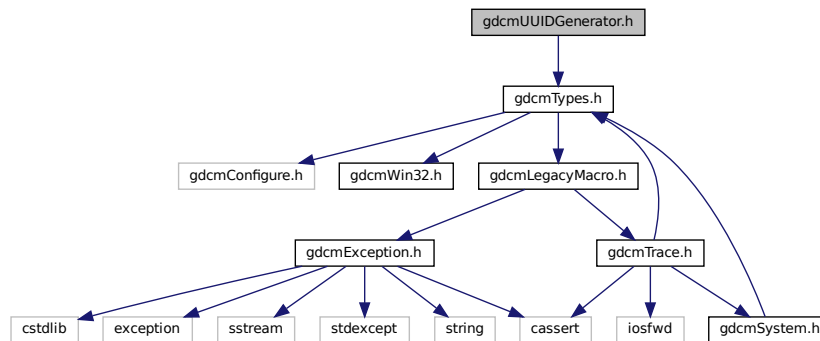
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 11.281 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



## Classes

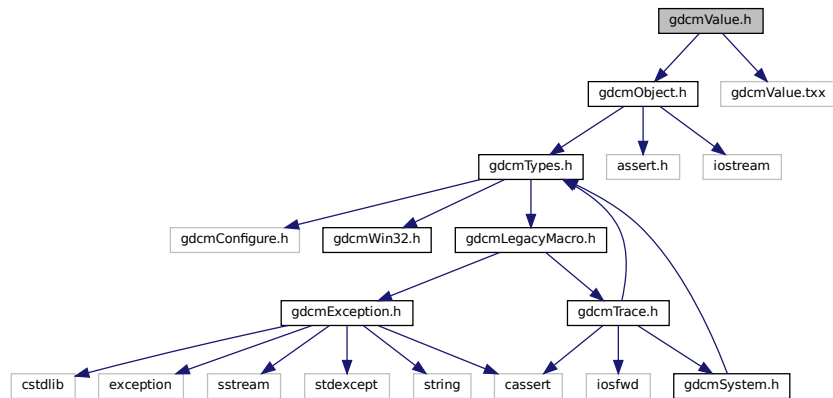
- class [gdcm::UUIDGenerator](#)  
*Class for generating unique UUID.*

## Namespaces

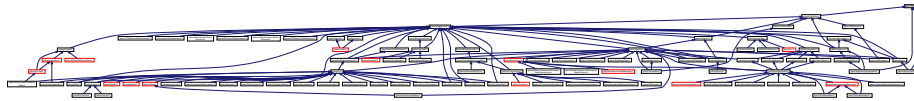
- [gdcm](#)



Include dependency graph for `gdcmValue.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

- [gdcm](#)

## 11.284 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

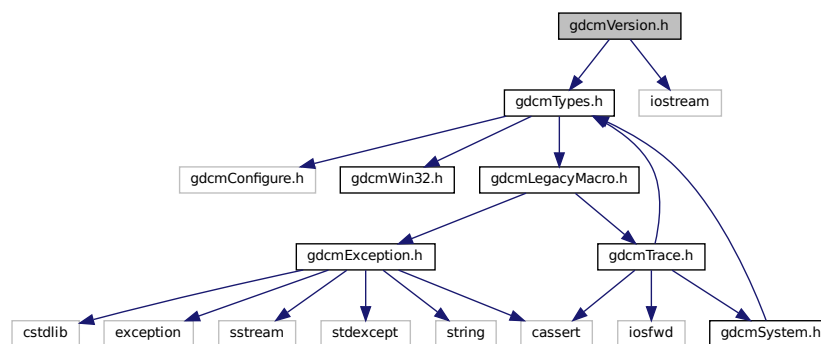
- [gdcm](#)

## 11.285 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

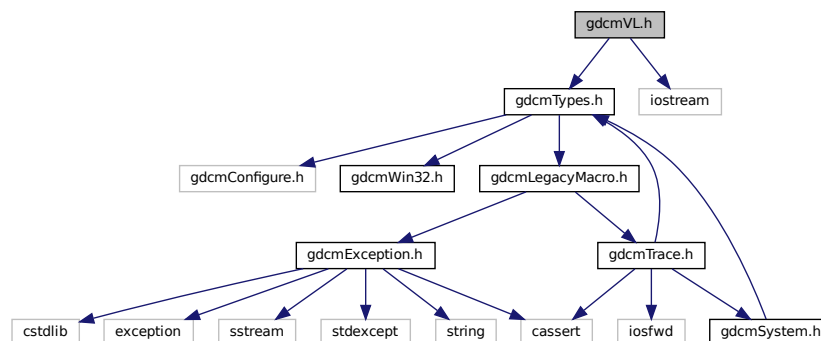
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

## 11.286 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

- [gdcm](#)

## Functions

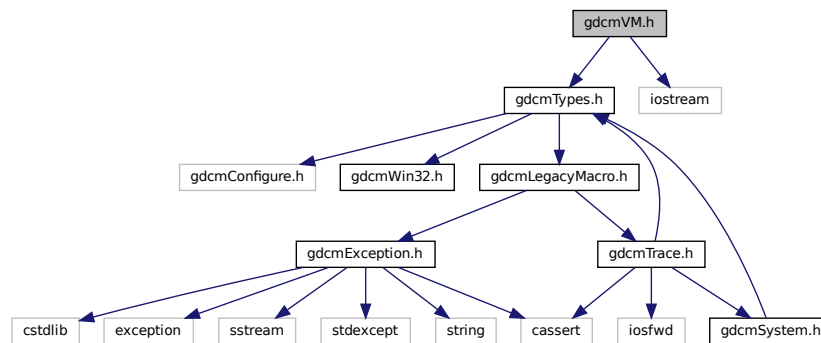
- `std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)`

## 11.287 gdcmVM.h File Reference

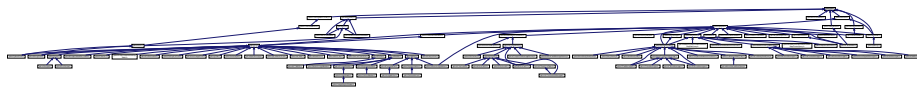
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmVM.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::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 [gdcm::VMToLength< T >](#)

## Namespaces

- [gdcm](#)

## Macros

- `#define TYPETOLENGTH(type, length)`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)`

## 11.287.1 Macro Definition Documentation

### 11.287.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(  
    type,  
    length )
```

#### Value:

```
template<> struct VMToLength<VM::type> \  
{ enum { Length = length }; };
```

## 11.288 gdcmVR.h File Reference

```
#include "gdcmTag.h"  
#include "gdcmTrace.h"  
#include "gdcmString.h"  
#include <iostream>  
#include <fstream>  
#include <assert.h>
```

Include dependency graph for gdcmVR.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [gdcm::UI](#)
- class [gdcm::VR](#)  
    *VR class.*
- struct [gdcm::VRToEncoding< T >](#)
- struct [gdcm::VRToType< T >](#)

## Namespaces

- [gdcm](#)

## Macros

- #define [TYPETOENCODING](#)(type, rep, rtype)
- #define [VRTypeTemplateCase](#)(type)

## Typedefs

- typedef String<'\', 16 > [gdcm::AECComp](#)
- typedef String<'\', 64 > [gdcm::ASComp](#)
- typedef String<'\', 16 > [gdcm::CSCComp](#)
- typedef String<'\', 64 > [gdcm::DACComp](#)
- 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<'\', 64, 0 > [gdcm::UIComp](#)
- typedef String<'\', 64 > [gdcm::UTComp](#)

## Functions

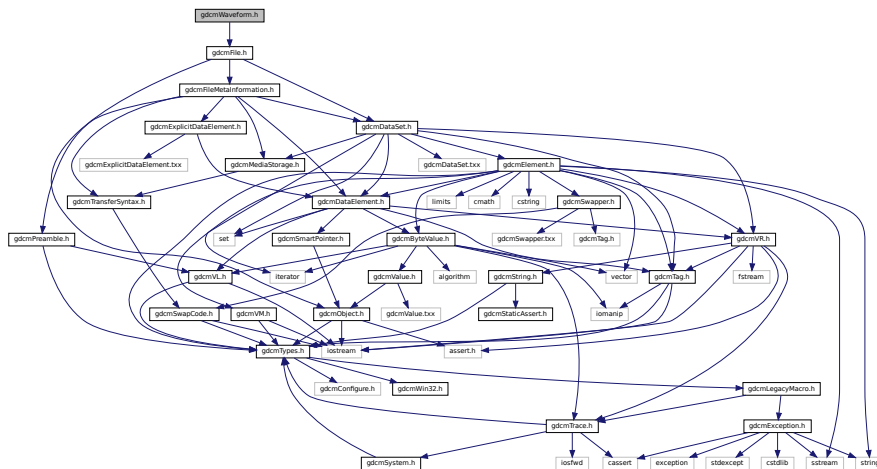
- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const VR &val)
- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const UI &\_val)
- [gdcm::TYPETOENCODING](#) (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN



- class `gdcm::VR16ExplicitDataElement`  
*Class to read/write a `DataElement` as Explicit Data `Element`.*

- **gdcm**

```
#include "gdcmFile.h"
Include dependency graph for gdcmWaveform.h:
```



- class `gdcm::Waveform`  
*Waveform* class.

- `gdcm`

This graph shows which files directly or indirectly include this file:



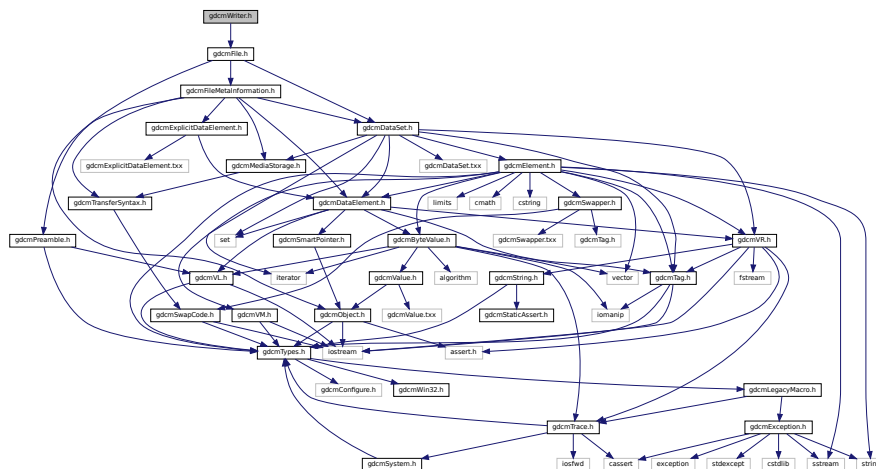




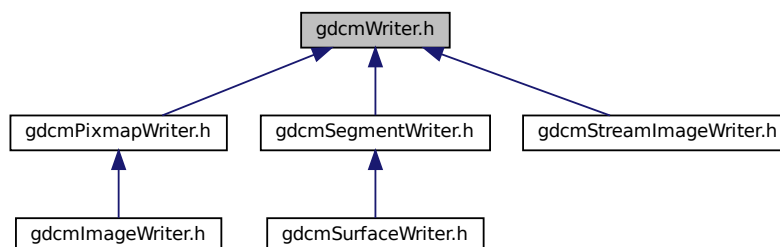
## 11.293 gdcmWriter.h File Reference

```
#include "gdcmFile.h"
```

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

- [gdcm](#)



## Classes

- class [gdcm::XMLPrinter](#)

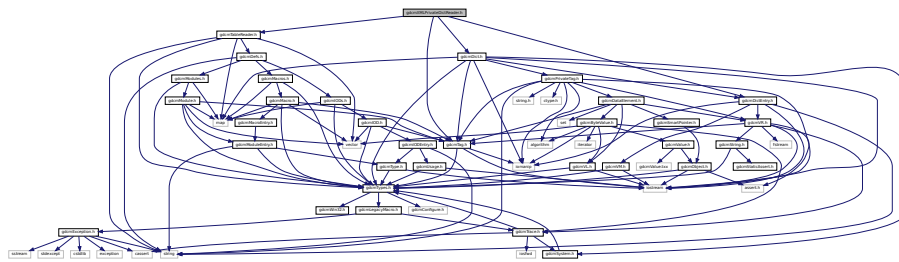
## Namespaces

- [gdcm](#)

## 11.296 gdcmXMLPrivateDictReader.h File Reference

```
#include "gdcmTableReader.h"  
#include "gdcmDict.h"  
#include "gdcmDictEntry.h"  
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLPrivateDictReader.h:



## Classes

- class [gdcm::XMLPrivateDictReader](#)  
*Class for representing a [XMLPrivateDictReader](#).*

## Namespaces

- [gdcm](#)

## 11.297 README.txt File Reference

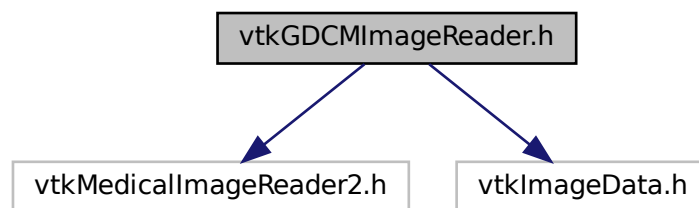
## 11.298 TestsList.txt File Reference

## 11.299 vtkGDCMImageReader.h File Reference

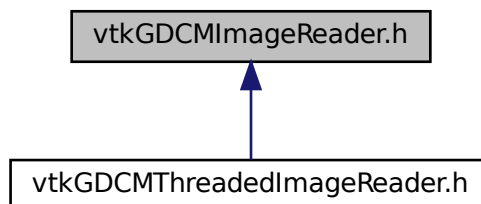
```
#include "vtkMedicalImageReader2.h"
```

```
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [vtkGDCMImageReader](#)

### Namespaces

- [gdcm](#)

## Macros

- #define [VTK\\_CMYK](#) 8
- #define [VTK\\_INVERSE\\_LUMINANCE](#) 5
- #define [VTK\\_LOOKUP\\_TABLE](#) 6
- #define [VTK\\_YBR](#) 7

### 11.299.1 Macro Definition Documentation

#### 11.299.1.1 VTK\_CMYK

```
#define VTK_CMYK 8
```

#### 11.299.1.2 VTK\_INVERSE\_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

#### 11.299.1.3 VTK\_LOOKUP\_TABLE

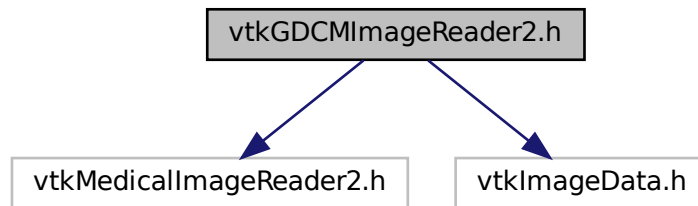
```
#define VTK_LOOKUP_TABLE 6
```

#### 11.299.1.4 VTK\_YBR

```
#define VTK_YBR 7
```

## 11.300 vtkGDCMImageReader2.h File Reference

```
#include "vtkMedicalImageReader2.h"  
#include "vtkImageData.h"  
Include dependency graph for vtkGDCMImageReader2.h:
```



### Classes

- class [vtkGDCMImageReader2](#)

### Namespaces

- [gdc](#)

### Macros

- `#define VTK_CMYK 8`
- `#define VTK_INVERSE_LUMINANCE 5`
- `#define VTK_LOOKUP_TABLE 6`
- `#define VTK_YBR 7`

### 11.300.1 Macro Definition Documentation

#### 11.300.1.1 VTK\_CMYK

```
#define VTK_CMYK 8
```

### 11.300.1.2 VTK\_INVERSE\_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

### 11.300.1.3 VTK\_LOOKUP\_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

### 11.300.1.4 VTK\_YBR

```
#define VTK_YBR 7
```

## 11.301 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



## Classes

- class [vtkGDCMImageWriter](#)

### 11.302 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



#### Classes

- class [vtkGDCMMedicalImageProperties](#)

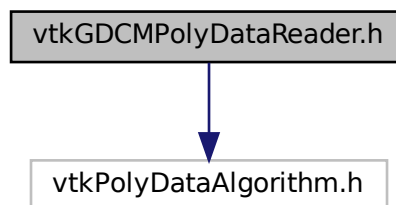
#### Namespaces

- [gdcm](#)

### 11.303 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:





## Classes

- class [vtkGDCMPolyDataReader](#)

## Namespaces

- [gdc](#)

## 11.304 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"
Include dependency graph for vtkGDCMPolyDataWriter.h:
```



## Classes

- class [vtkGDCMPolyDataWriter](#)

## Namespaces

- [gdc](#)

### 11.305 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



#### Classes

- class [vtkGDCMTesting](#)

### 11.306 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



## Classes

- class [vtkGDCMThreadedImageReader](#)

## 11.307 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



## Classes

- class [vtkGDCMThreadedImageReader2](#)

## 11.308 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkImageColorViewer.h:



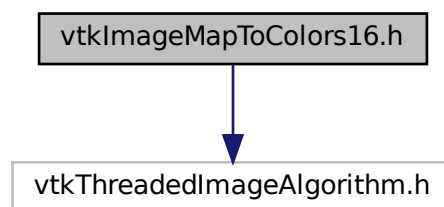
## Classes

- class [vtkImageColorViewer](#)

## 11.309 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



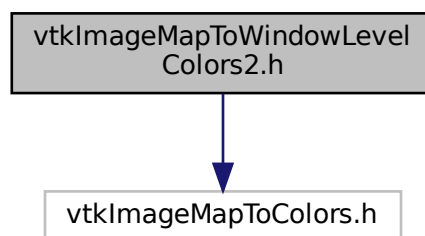
## Classes

- class [vtkImageMapToColors16](#)

## 11.310 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



## Classes

- class [vtkImageMapToWindowLevelColors2](#)

## 11.311 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



## Classes

- class [vtkImagePlanarComponentsToComponents](#)

## 11.312 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



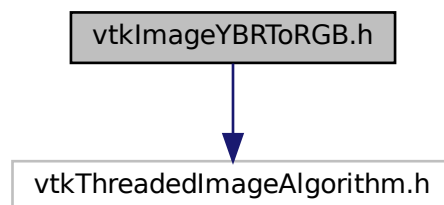
## Classes

- class [vtkImageRGBToYBR](#)

### 11.313 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



## Classes

- class [vtkImageYBRToRGB](#)

### 11.314 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
```

```
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



## Classes

- class [vtkLookupTable16](#)

## 11.315 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



## Classes

- class [vtkRTStructSetProperties](#)





## Chapter 12

# Example Documentation

### 12.1 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.GetFileNames());

        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 theFileNames =
    theScanner.GetAllFileNamesFromTagToValue(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(theFileNames);
FileNamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
String studyUID = theScanner.GetValue(theScanner.GetFileNames().get(0), theStudyTag);
long theNumSeries = theSeriesValues.size();
for (int i = 0; i < theNumSeries; i++) {
    FileNamesType theSeriesFiles =
        theScanner.GetAllFileNamesFromTagToValue(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 creatin 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 saggitalColors = new vtkImageMapToColors();
saggitalColors.SetInput(theImageData);
saggitalColors.SetLookupTable(bwLut);
vtkImageActor saggital = new vtkImageActor();
saggital.SetInput(saggitalColors.GetOutput());
saggital.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(saggital);
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/mmroden/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);
}
}

```

## 12.2 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;
    }
}

```

## 12.3 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;
}

```

## 12.4 CastConvertPhilips.py

```

1
14
15 """
16 Usage:
17
18 python --public /path/to/directory/
19 or
20 python --private /path/to/directory/
21
22 python --public --extension bak /path/to/directory/
23
24 rename -f 's/\.bak$//' *.bak
25
26 TODO:
27 http://docs.python.org/library/optparse.html#module-optparse
28 """
29
30 import vtkgdc
31 import vtk
32 import sys
33 import gdc
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36     gdc.ImageHelper.SetForceRescaleInterceptSlope(True)
37     vtkreader = vtkgdc.vtkGDCMImageReader()
38     vtkreader.SetFileName( filename )
39     vtkreader.Update()
40
41     cast = vtk.vtkImageCast()
42     cast.SetInput( vtkreader.GetOutput() )
43     cast.SetOutputScalarTypeToUnsignedShort()
44
45     # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
46     # Some operation will actually be discarded (we simply need a temp storage)
47     vtkwriter = vtkgdc.vtkGDCMImageWriter()
48     vtkwriter.SetFileName( tmpfile )
49     vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
50     vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
51     print "Format:", vtkreader.GetImageFormat()
52     vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
53     vtkwriter.SetInput( cast.GetOutput() )
54     #vtkwriter.Update()
55     vtkwriter.Write()
56
57     # ok now rewrite the exact same file as the original (keep all info)
58     # but use the Pixel Data Element from the written file
59     tmpreader = gdc.ImageReader()
60     tmpreader.SetFileName( tmpfile )
61     if not tmpreader.Read():
62         sys.exit(1)
63
64     reader = gdc.Reader()
65     reader.SetFileName( filename )
66     if not reader.Read():
67         sys.exit(1)
68
69     # Make sure to remove Slope/Rescale to avoid re-execution
70     ds = reader.GetFile().GetDataSet()
71     tags = [
72         gdc.Tag(0x0028,0x1052),
73         gdc.Tag(0x0028,0x1053),
74         gdc.Tag(0x0028,0x1053),
75     ]
76     for tag in tags:
77         ds.Remove( tag )
78
79     writer = gdc.ImageWriter()
80     writer.SetFileName( outfilename )
81     # Pass image from vtk written file

```



```

82  writer.SetImage( tmpreader.GetImage() )
83  # pass dataset from initial 'reader'
84  writer.SetFile( reader.GetFile() )
85  if not writer.Write():
86      sys.exit(1)
87
88  def ProcessOneFilePrivate(filename, outfilename, tmpfile):
89      vtkreader = vtkgdcmm.vtkGDCMImageReader()
90      vtkreader.SetFileName( filename )
91      vtkreader.Update()
92
93
94      # (2005,1409)      DS      4      0.0
95      # (2005,140a)      DS      16     1.52283272283272
96
97      # (2005,0014)      LO      26     Philips MR Imaging DD 005
98      tag1 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
99      tag2 = gdcmm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
100
101
102
103      # Need to access some private tags, reread the file (for now):
104      reader = gdcmm.Reader()
105      reader.SetFileName( filename )
106      if not reader.Read():
107          sys.exit(1)
108
109      ds = reader.GetFile().GetDataSet()
110
111      e11 = ds.GetDataElement( tag1 )
112      e12 = ds.GetDataElement( tag2 )
113
114
115      #pf = gdcmm.PythonFilter()
116      #pf.SetFile( reader.GetFile() )
117      #print e11.GetTag()
118
119      print e11.GetByteValue()
120      v1 = eval(e11.GetByteValue().GetBuffer())
121      print e12.GetByteValue()
122      v2 = eval(e12.GetByteValue().GetBuffer())
123
124      print v1
125      shift = v1
126      print v2
127      scale = v2
128
129      ss = vtk.vtkImageShiftScale()
130      ss.SetInput( vtkreader.GetOutput() )
131      # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
132      assert shift == 0
133      ss.SetShift( shift )
134      ss.SetScale( scale )
135      ss.SetOutputScalarTypeToUnsignedShort()
136      ss.Update()
137
138      # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
139      # Some operation will actually be discarded (we simply need a temp storage)
140      vtkwriter = vtkgdcmm.vtkGDCMImageWriter()
141      vtkwriter.SetFileName( tmpfile )
142      vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
143      vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
144      vtkwriter.SetImageFormat( reader.GetImageFormat() )
145      # do not pass shift/scale again
146      vtkwriter.SetInput( ss.GetOutput() )
147      #vtkwriter.Update()
148      vtkwriter.Write()
149
150      # ok now rewrite the exact same file as the original (keep all info)
151      # but use the Pixel Data Element from the written file
152      tmpreader = gdcmm.ImageReader()
153      tmpreader.SetFileName( tmpfile )
154      if not tmpreader.Read():
155          sys.exit(1)
156
157      writer = gdcmm.ImageWriter()
158      writer.SetFileName( outfilename )
159      # Pass image from vtk written file
160      writer.SetImage( tmpreader.GetImage() )
161      # pass dataset from initial 'reader'
162      writer.SetFile( reader.GetFile() )

```

```

163     if not writer.Write():
164         sys.exit(1)
165
166 if __name__ == "__main__":
167
168     gdcm.Trace.DebugOff()
169     gdcm.Trace.WarningOff()
170     #filename = sys.argv[1]
171     #outfilename = sys.argv[2]
172     tmpfile = "/tmp/philips_rescaled.dcm"
173     #ProcessOneFile( filename, outfilename, tmpfile )
174     rescaletype = sys.argv[1]
175     assert rescaletype == "--public" or rescaletype == "--private"
176     dirname = sys.argv[2]
177     d = gdcm.Directory()
178     d.Load( dirname )
179
180     for f in d.GetFilenames():
181         #print f
182         ProcessOneFilePublic( f, f + ".bak", tmpfile )
183
184
185 print "success"

```

## 12.5 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;
gdcmm::DataElement del = ds.GetDataElement( pt1 ); // Convert Private tag,
    into actual DataElement
std::cout << del << std::endl;
del.SetByteValue( str1, (uint32_t)strlen(str1) );
ds.Replace( del );

const char str2[] = "GDCM was here 2!";
if( !ds.FindDataElement( pt2 ) ) return 1;
gdcmm::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;
gdcmm::DataElement de3 = ds.GetDataElement( pt3 );
std::cout << de3 << std::endl;
de3.SetByteValue( str3, (uint32_t)strlen(str3) );
ds.Replace( de3 );

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.6 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 << 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::Tag tsis(0x0008,0x2112); // SourceImageSequence
    if ( ds.FindDataElement( tsis ) )
    {
        const gdcmm::DataElement &sis = ds.GetDataElement( tsis );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqsis = sis.
            GetValueAsSQ();
        if ( sqsis && sqsis->GetNumberOfItems() )
        {
            gdcmm::Item &item1 = sqsis->GetItem(1);
            gdcmm::DataSet &nestedds = item1.GetNestedDataSet();
            gdcmm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
            if( nestedds.FindDataElement( tprcs ) )
            {
                const gdcmm::DataElement &prcs = nestedds.GetDataElement( tprcs );
                gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqprcs = prcs.
                    GetValueAsSQ();
                if ( sqprcs && sqprcs->GetNumberOfItems() )
                {
                    gdcmm::Item &item2 = sqprcs->GetItem(1);
                    gdcmm::DataSet &nestedds2 = item2.GetNestedDataSet();
                    // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
                    gdcmm::Tag tcm(0x0008,0x0104);
                    if( nestedds2.FindDataElement( tcm ) )
                    {
                        gdcmm::DataElement cm = nestedds2.GetDataElement( tcm );
                        std::string mystr = "GDCM was here";
                        cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                        nestedds2.Replace( cm );
                    }
                }
            }
        }
    }

    gdcmm::Writer writer;
    writer.SetFile( file );
    writer.SetFileName( outfilename );
    if ( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.7 CheckBigEndianBug.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.

=====*/
/*
 * 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 vbindiff 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 "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.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];

    gdcm::ImageReader reader1;
    reader1.SetFileName( filename1 );
    if( !reader1.Read() )
    {
        std::cerr << "Could not read: " << filename1 << std::endl;
        return 1;
    }

    gdcm::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 = gdcm::System::FileSize(filename1);
    size_t s2 = gdcm::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);

    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;
}

```

## 12.8 ClinicalTrialAnnotate.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 implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 * ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.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( gdcmm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
// <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
ano.Replace( gdcmm::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( gdcmm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );

gdcmm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}
```

## 12.9 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://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.
=====*/

/*
 * 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)
{
    gdcml.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";
    gdcml.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcml.UIDGenerator.
        GetRoot() );

    gdcml.Global global = gdcml.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( !gdcml.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcml.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.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( ano , filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
    }
}

return 0;
}
}

```

## 12.10 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();

    const gdcm::Image &image = reader.GetImage();
    image.Print( std::cout );

    gdcm::ImageChangeTransferSyntax change;
    change.SetTransferSyntax(

```

```

        gdcmm::TransferSyntax::JPEG2000Lossless );
change.SetTransferSyntax(
    gdcmm::TransferSyntax::JPEGLosslessProcess14_1 );
//change.SetTransferSyntax( gdcmm::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( outfilename, std::ios::binary );
//image.GetBuffer2(out);
//out.close();
gdcmm::ImageWriter writer;
writer.SetImage( change.GetOutput() );
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 12.11 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.TType.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;
}
}

```

## 12.12 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 "gdcmIPPSorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType double
#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 ippszspacing = 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();
    #if (VTK_MAJOR_VERSION >= 6)
        v16->SetInputConnection( reader->GetOutputPort() );
    #else
        v16->SetInput( reader->GetOutput() );
    #endif
    v16->SetOutputSpacing( spacing[0], spacing[1], ippszspacing );
    v16->Update();

    v16->GetOutput()->Print( std::cout );

    return 0;
}

```

## 12.13 Convert16BitsTo8Bits.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 "gdcmtesting.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:
// gdcmtesting/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcmtesting::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();
    #if (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" );
    #if (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;
}

```

## 12.14 ConvertMPL.py

```

1
14
15 """
16 display a DICOM image with matplotlib via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
27 """

```

```

28
29 import gdc
30 import numpy
31 from pylab import *
32
33
34 def get_gdcm_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
37                 gdcm.PixelFormat.INT8 :numpy.uint8,
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
39                 gdcm.PixelFormat.INT16 :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32 :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32:numpy.float32,
43                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52     """
53     pf = image.GetPixelFormat().GetScalarType()
54     print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57         "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size: %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64
65     result.shape = d
66     return result
67
68
69 if __name__ == "__main__":
70     import sys
71     r = gdc.ImageReader()
72     filename = sys.argv[1]
73     r.SetFileName( filename )
74     if not r.Read(): sys.exit(1)
75     numpy_array = gdcm_to_numpy( r.GetImage() )
76
77     subplot(111)# one plot, on left
78     title(filename)
79
80     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
81
82     subplots_adjust(bottom=0.1, right=0.8, top=0.9)
83     cax = axes([0.85, 0.1, 0.075, 0.8])
84     colorbar(cax=cax)
85     title('values')
86     get_current_fig_manager().window.title('plot')
87     show()

```

## 12.15 ConvertMultiFrameToSingleFrame.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 "vtkStringArray.h"

#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"

int main(int argc, char *argv[])
{
    std::string filename;
    if( argc <= 1 )
    {
        const char *directory = gdcm::Testing::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";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = dims[2];
    fg.SetNumberOfFileNames( 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 );
    #if (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;
}

```



## 12.16 ConvertNumpy.py

```

1
14
15 """
16 This module add support for converting a gdcm.Image to a numpy array.
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Removed:
22 - float16 is defined in GDCM API but no implementation exist for it ...
23 """
24
25 import gdcm
26 import numpy
27
28 def get_gdcm_to_numpy_typemap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.uint8,
31                 gdcm.PixelFormat.INT8 :numpy.int8,
32                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
33                 #gdcm.PixelFormat.INT12 :numpy.int12,
34                 gdcm.PixelFormat.UINT16 :numpy.uint16,
35                 gdcm.PixelFormat.INT16 :numpy.int16,
36                 gdcm.PixelFormat.UINT32 :numpy.uint32,
37                 gdcm.PixelFormat.INT32 :numpy.int32,
38                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
39                 gdcm.PixelFormat.FLOAT32:numpy.float32,
40                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
41     return _gdcm_np
42
43 def get_numpy_array_type(gdcm_pixel_format):
44     """Returns a numpy array typecode given a GDCM Pixel Format."""
45     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
46
47 def gdcm_to_numpy(image):
48     """Converts a GDCM image to a numpy array.
49     """
50     pf = image.GetPixelFormat()
51
52     assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
53         "Unsupported array type %s"%pf
54
55     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
56     if image.GetNumberOfDimensions() == 3:
57         shape = shape[0] * image.GetDimension(2), shape[1]
58
59     dtype = get_numpy_array_type(pf.GetScalarType())
60     gdcm_array = image.GetBuffer()
61     result = numpy.frombuffer(gdcm_array, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66     import sys
67     r = gdcm.ImageReader()
68     filename = sys.argv[1]
69     r.SetFileName( filename )
70     if not r.Read():
71         sys.exit(1)
72
73     numpy_array = gdcm_to_numpy( r.GetImage() )
74     print numpy_array

```

## 12.17 ConvertPIL.py

```

1
14
15 """
16 save a DICOM image with PIL via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12

```

```

20
21 Usage:
22
23 python ConvertNumpy.py "IM000000"
24
25 Thanks:
26 plotting example - Ray Schumacher 2009
27 """
28
29 import gdcm
30 import numpy
31 from PIL import Image, ImageOps
32
33
34 def get_gdcm_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
37                 gdcm.PixelFormat.INT8 :numpy.uint8,
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
39                 gdcm.PixelFormat.INT16 :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32 :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32 :numpy.float32,
43                 gdcm.PixelFormat.FLOAT64 :numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52     """
53     pf = image.GetPixelFormat().GetScalarType()
54     print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57         "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size: %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62     result = numpy.frombuffer(gdcm_array, dtype=dtype)
63     maxV = float(result[result.argmax()])
64
65     result = numpy.log(result+50)
66     maxV = float(result[result.argmax()])
67     result = result*(2.**8/maxV)
68     result.shape = d
69     return result
70
71
72
73 if __name__ == "__main__":
74     import sys
75     r = gdcm.ImageReader()
76     filename = sys.argv[1]
77     r.SetFileName( filename )
78     if not r.Read(): sys.exit(1)
79     numpy_array = gdcm_to_numpy( r.GetImage() )
80
81     pilImage = Image.frombuffer('L',
82                                numpy_array.shape,
83                                numpy_array.astype(numpy.uint8),
84                                'raw','L',0,1)
85
86
87     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
88     pilImage.save(sys.argv[1]+' .jpg')

```

## 12.18 ConvertRGBToLuminance.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 "vtkImageLuminance.h"

#include "gdcmTesting.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 = gdcm::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();
    #if (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;
}

```

## 12.19 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"

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( outfile );
    //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;
}

```

## 12.20 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)
{
    const 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:
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubbuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        buffer16++;
    }

    QImage Qt = 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;
}

```

## 12.21 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.rgba 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;
}

```

## 12.22 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;
}
```

## 12.23 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 "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 *[])
{
    gdcm::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";
    gdcm::FilenameGenerator fg;
    fg.SetPattern( os.str().c_str() );
    unsigned int nfiles = zSize;
    fg.SetNumberOfFilenames( 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;
    }

    vtkStringArray *filenames = vtkStringArray::New();
    for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++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;
}

```

## 12.24 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 "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 automatically 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;
}

```

## 12.25 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];

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( outfile );

    file.GetHeader().SetDataSetTransferSyntax(
        gdcm::TransferSyntax::JPIPReferenced );

    gdcm::Anonymizer anon;
    anon.SetFile( file );

    gdcm::MediaStorage ms =
        gdcm::MediaStorage::SecondaryCaptureImageStorage;

    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() );
    //
    anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
    anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
    anon.Empty( gdcm::Tag(0x0010,0x30) );
    anon.Empty( gdcm::Tag(0x0010,0x40) );
    anon.Empty( gdcm::Tag(0x0008,0x20) );
    anon.Empty( gdcm::Tag(0x0008,0x30) );
    anon.Empty( gdcm::Tag(0x0008,0x90) );
    anon.Empty( gdcm::Tag(0x0020,0x10) );
    anon.Empty( gdcm::Tag(0x0020,0x11) );
    anon.Empty( gdcm::Tag(0x0008,0x50) );
    anon.Empty( gdcm::Tag(0x0020,0x0013) );
    anon.Replace( gdcm::Tag(0x0020,0xd), gen.Generate() );
    anon.Replace( gdcm::Tag(0x0020,0xe), gen.Generate() );
    anon.Replace( gdcm::Tag(0x0008,0x64), "WSD " );
    anon.Replace( gdcm::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: " << outfilename << std::endl;
    return 1;
}

return 0;
}

```

## 12.26 CreateRAWStorage.py

```

1
14
15 """
16 <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"/>
17 """
18
19 import gdcmm
20 import sys,os
21
22 if __name__ == "__main__":
23     r = gdcmm.Reader()
24     # Will require Testing...
25     dataroot = gdcmm.Testing.GetDataRoot()
26     filename = os.path.join( dataroot, '012345.002.050.dcm' )
27     r.SetFileName( filename )
28     r.Read()
29     f = r.GetFile()
30     ds = f.GetDataSet()
31
32     uid = "1.2.840.10008.5.1.4.1.1.66"
33     # f = gdcmm.File()
34     # ds = f.GetDataSet()
35     de = gdcmm.DataElement( gdcmm.Tag(0x0008,0x0016) )
36     de.SetByteValue( uid, gdcmm.VL(len(uid)) )
37     vr = gdcmm.VR( gdcmm.VR.UI )
38     de.SetVR( vr )
39     ds.Replace( de )
40
41     ano = gdcmm.Anonymizer()
42     ano.SetFile( r.GetFile() )
43     ano.RemovePrivateTags()
44     ano.RemoveGroupLength()
45     taglist = [
46         gdcmm.Tag(0x0008,0x0008),
47         gdcmm.Tag(0x0008,0x0022),
48         gdcmm.Tag(0x0008,0x0032),
49         gdcmm.Tag(0x0008,0x2111),
50         gdcmm.Tag(0x0008,0x1150),
51         gdcmm.Tag(0x0008,0x1155),
52         gdcmm.Tag(0x0008,0x0100),
53         gdcmm.Tag(0x0008,0x0102),
54         gdcmm.Tag(0x0008,0x0104),
55         gdcmm.Tag(0x0040,0xa170),
56         gdcmm.Tag(0x0008,0x2112),
57         gdcmm.Tag(0x0008,0x0100),
58         gdcmm.Tag(0x0008,0x0102),
59         gdcmm.Tag(0x0008,0x0104),
60         gdcmm.Tag(0x0008,0x9215),
61         gdcmm.Tag(0x0018,0x0010),
62         gdcmm.Tag(0x0018,0x0022),
63         gdcmm.Tag(0x0018,0x0050),
64         gdcmm.Tag(0x0018,0x0060),
65         gdcmm.Tag(0x0018,0x0088),
66         gdcmm.Tag(0x0018,0x0090),
67         gdcmm.Tag(0x0018,0x1040),
68         gdcmm.Tag(0x0018,0x1100),

```

```

69  gdcm.Tag(0x0018,0x1110),
70  gdcm.Tag(0x0018,0x1111),
71  gdcm.Tag(0x0018,0x1120),
72  gdcm.Tag(0x0018,0x1130),
73  gdcm.Tag(0x0018,0x1150),
74  gdcm.Tag(0x0018,0x1151),
75  gdcm.Tag(0x0018,0x1152),
76  gdcm.Tag(0x0018,0x1160),
77  gdcm.Tag(0x0018,0x1190),
78  gdcm.Tag(0x0018,0x1210),
79  gdcm.Tag(0x0020,0x0012),
80  gdcm.Tag(0x0020,0x0032),
81  gdcm.Tag(0x0020,0x0037),
82  gdcm.Tag(0x0020,0x1041),
83  gdcm.Tag(0x0020,0x4000),
84  gdcm.Tag(0x0028,0x0002),
85  gdcm.Tag(0x0028,0x0004),
86  gdcm.Tag(0x0028,0x0010),
87  gdcm.Tag(0x0028,0x0011),
88  gdcm.Tag(0x0028,0x0030),
89  gdcm.Tag(0x0028,0x0100),
90  gdcm.Tag(0x0028,0x0101),
91  gdcm.Tag(0x0028,0x0102),
92  gdcm.Tag(0x0028,0x0103),
93  gdcm.Tag(0x0028,0x1052),
94  gdcm.Tag(0x0028,0x1053),
95  gdcm.Tag(0x0028,0x2110),
96  gdcm.Tag(0x0028,0x2112),
97  gdcm.Tag(0x7fe0,0x0010),
98  gdcm.Tag(0x0018,0x0020),
99  gdcm.Tag(0x0018,0x0021),
100 gdcm.Tag(0x0018,0x0023),
101 gdcm.Tag(0x0018,0x0025),
102 gdcm.Tag(0x0018,0x0080),
103 gdcm.Tag(0x0018,0x0081),
104 gdcm.Tag(0x0018,0x0083),
105 gdcm.Tag(0x0018,0x0084),
106 gdcm.Tag(0x0018,0x0085),
107 gdcm.Tag(0x0018,0x0086),
108 gdcm.Tag(0x0018,0x0087),
109 gdcm.Tag(0x0018,0x0091),
110 gdcm.Tag(0x0018,0x0093),
111 gdcm.Tag(0x0018,0x0094),
112 gdcm.Tag(0x0018,0x0095),
113 gdcm.Tag(0x0018,0x1088),
114 gdcm.Tag(0x0018,0x1090),
115 gdcm.Tag(0x0018,0x1094),
116 gdcm.Tag(0x0018,0x1250),
117 gdcm.Tag(0x0018,0x1251),
118 gdcm.Tag(0x0018,0x1310),
119 gdcm.Tag(0x0018,0x1312),
120 gdcm.Tag(0x0018,0x1314),
121 gdcm.Tag(0x0018,0x1315),
122 gdcm.Tag(0x0018,0x1316),
123 gdcm.Tag(0x0020,0x0110),
124 gdcm.Tag(0x0028,0x0120),
125 gdcm.Tag(0x0028,0x1050),
126 gdcm.Tag(0x0028,0x1051)
127 ]
128 for tag in taglist:
129     #print tag
130     ano.Remove( tag )
131
132 # special handling
133 gen = gdcm.UIDGenerator()
134 ano.Replace( gdcm.Tag(0x0008,0x9123), gen.Generate() )
135 #ano.Empty( gdcm.Tag(0x0040,0x0555) )
136
137
138 #
139 # uid = gen.Generate()
140 # de.SetTag( gdcm.Tag(0x0008,0x0018) )
141 # de.SetByteValue( uid, gdcm.VL(len(uid)) )
142 # ds.Insert( de )
143
144 # init FMI now:
145 #fmi = f.GetHeader()
146 #ts = gdcm.TransferSyntax()
147 #print ts
148 #fmi.SetDataSetTransferSyntax( ts ) # default
149 #print fmi.GetDataSetTransferSyntax()

```

```

150 #de.SetTag( gdcm.Tag(0x0002,0x0010) )
151 #uid = "1.2.840.10008.1.2"
152 #de.SetByteValue( uid, gdcm.VL(len(uid)) )
153 #fmi.Insert( de )
154 # f.SetHeader( r.GetFile().GetHeader() )
155
156 writer = gdcm.Writer()
157 writer.SetFile( ano.GetFile() )
158 writer.SetFileName( "rawstorage.dcm" );
159 writer.Write()

```

## 12.27 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;
    }

    gdcm::CSAHeader csa;
    const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

    const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
    //std::cout << t1 << std::endl;
    //const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

    if( ds.FindDataElement( t1 ) )
    {
        csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
    }
}

```

```

    csa.Print( std::cout );
}
int dims[2] = {};
if( csa.FindCSAElementByName( "Columns" ) )
{
    const gdcm::CSAElement &csael = csa.GetCSAElementByName( "Columns" );
    ;
    std::cout << csael << std::endl;
    //const gdcm::ByteValue *bv = csael.GetByteValue();
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el;
    el.Set( csael.GetValue() );
    dims[0] = el.GetValue();
    std::cout << "Columns:" << el.GetValue() << std::endl;
}

if( csa.FindCSAElementByName( "Rows" ) )
{
    const gdcm::CSAElement &csael2 = csa.GetCSAElementByName( "Rows" );
    std::cout << csael2 << std::endl;
    gdcm::Element<gdcm::VR::IS, gdcm::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 gdcm::CSAElement &csael3 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !csael3.IsEmpty() )
    {
        std::cout << csael3 << std::endl;
        gdcm::Element<gdcm::VR::DS, gdcm::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;
}

gdcm::ImageWriter writer;

gdcm::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] );
gdcm::PixelFormat pixeltype = gdcm::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() );

gdcm::PhotometricInterpretation pi;
pi = gdcm::PhotometricInterpretation::MONOCHROME2;
//pixeltype.SetSamplesPerPixel( );
image.SetPhotometricInterpretation( pi );
image.SetPixelFormat( pixeltype );
//image.SetIntercept( inputimage.GetIntercept() );
//image.SetSlope( inputimage.GetSlope() );

//gdcm::DataElement pixeldata( gdcm::Tag(0x7fe1,0x1010) );

```



```

//pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
gdcm::PrivateTag csanonimaget(0x7fel,0x10,"SIEMENS CSA NON-IMAGE");
const gdcm::DataElement &pixeldata = ds.GetDataElement( csanonimaget );
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;
}

```

## 12.28 CStoreQtProgress.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 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 gdcmData/MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcm {
/*
 * 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++;
        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 gdcm

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;
    gdcm::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);

    gdcm::SmartPointer<gdcm::ServiceClassUser> scup = new
        gdcm::ServiceClassUser;
    gdcm::ServiceClassUser &scu = *scup;
    //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
    // let's use a more complicated progress reported in this example
    gdcm::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;
    }

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );

    // setup the PC(s) based on the filenames:
    gdcm::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();
}

```

## 12.29 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;
}

```

## 12.30 DecompressImage.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/DecompressImage.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressImage gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

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.TType.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 );
        }
    }
}

```

## 12.31 DecompressImage.py

```

1
14
15 """
16 Usage:
17
18 python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
19 """
20
21 import gdcm
22 import sys
23
24 if __name__ == "__main__":
25
26     file1 = sys.argv[1]
27     file2 = sys.argv[2]
28
29     r = gdcm.ImageReader()
30     r.SetFileName( file1 )

```

```

31  if not r.Read():
32      sys.exit(1)
33
34  # check GetFragment API:
35  pd = r.GetFile().GetDataSet().GetDataElement(gdcm.Tag(0x7fe0, 0x0010))
36  frags = pd.GetSequenceOfFragments();
37  frags.GetFragment(0);
38
39  ir = r.GetImage()
40  w = gdcm.ImageWriter()
41  image = w.GetImage()
42
43  image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
44  dims = ir.GetDimensions();
45  print ir.GetDimension(0);
46  print ir.GetDimension(1);
47  print "Dims:", dims
48
49  # Just for fun:
50  dircos = ir.GetDirectionCosines()
51  t = gdcm.Orientation.GetType(tuple(dircos))
52  l = gdcm.Orientation.GetLabel(t)
53  print "Orientation label:", l
54
55  image.SetDimension(0, ir.GetDimension(0) );
56  image.SetDimension(1, ir.GetDimension(1) );
57
58  pixeltype = ir.GetPixelFormat();
59  image.SetPixelFormat( pixeltype );
60
61  pi = ir.GetPhotometricInterpretation();
62  image.SetPhotometricInterpretation( pi );
63
64  pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
65  str1 = ir.GetBuffer()
66  #print ir.GetBufferLength()
67  pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
68  image.SetDataElement( pixeldata )
69
70  w.SetFileName( file2 )
71  w.SetFile( r.GetFile() )
72  w.SetImage( image )
73  if not w.Write():
74      sys.exit(1)

```

## 12.32 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 file are not guarantee 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;
}

```

## 12.33 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;
}
}

```

## 12.34 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.TType.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 );
    }
}
}

```

## 12.35 DeriveSeries.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 "gdcmAttribute.h"
#include "gdcmFileDerivation.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;
}

```

## 12.36 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;
}

```

## 12.37 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 && strncmp(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(), gdcm::t4 );
        assert( value );
        iopset.insert( value );
    }
    size_t n = iopset.size();
    if ( n == 0 )
    {
        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
        gdcm::DirectionCosines ref;
        gdcm::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(), gdcm::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 dirl;
    if( argc < 2 )
    {
        const char *extradataroot = NULL;

```

```

#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.c_str(), 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.GetFileNames() );
if( !b )
{
    std::cerr << "Scanner failed" << std::endl;
    return 1;
}

gdcml::DiscriminateVolume dv;
dv.ProcessIntoVolume( s );
dv.Print( std::cout );

return 0;
}

```

## 12.38 DumbAnonymizer.py

```

1
14
15 """
16 This example shows how one can use the gdcml.Anonymizer in 'dumb' mode.
17 This class becomes really handy when one knows which particular tag to fill in.
18
19 Usage:
20
21 python DumbAnonymizer.py gdcmlData/012345.002.050.dcm out.dcm
22
23 """
24
25 import gdcml
26
27 # http://www.oid-info.com/get/1.3.6.1.4.17434
28 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
29
30 tag_rules={
31     # Value
32     (0x0012,0x0010):("Value","MySponsorName"),
33     (0x0012,0x0020):("Value","MyProtocolID"),
34     (0x0012,0x0021):("Value","MyProtocolName"),
35     (0x0012,0x0062):("Value","YES"),
36     (0x0012,0x0063):("Value","MyDeidentificationMethod"),
37
38     # Method
39     (0x0002,0x0003):("Method","GenerateMSOPIId"),
40     (0x0008,0x1155):("Method","GenerateMSOPIId"),
41     (0x0008,0x0018):("Method","GenerateMSOPIId"),
42     (0x0010,0x0010):("Method","GetSponsorInitials"),
43     (0x0010,0x0020):("Method","GetSponsorId"),
44     (0x0012,0x0030):("Method","GetSiteId"),
45     (0x0012,0x0031):("Method","GetSiteName"),
46     (0x0012,0x0040):("Method","GetSponsorId"),
47     (0x0012,0x0050):("Method","GetTPIId"),
48     (0x0018,0x0022):("Method","KeepIfExist"),

```



```

49     (0x0018,0x1315):("Method","KeepIfExist"),
50     (0x0020,0x000d):("Method","GenerateStudyId"),
51     (0x0020,0x000e):("Method","GenerateSeriesId"),
52     (0x0020,0x1002):("Method","GetNumberOfFrames"),
53     (0x0020,0x0020):("Method","GetPatientOrientation"),
54     # Other:
55     (0x0012,0x0051):("Patient Field","Type Examen"),
56     (0x0018,0x1250):("Sequence Field","Receive Coil"),
57     (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
58     (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
59     (0x0018,0x0082):("Sequence Field","Inversion Time"),
60 }
61
62 class MyAnon:
63     def __init__(self):
64         self.studyuid = None
65         self.seriesuid = None
66         generator = gdcm.UIDGenerator()
67         if not self.studyuid:
68             self.studyuid = generator.Generate()
69         if not self.seriesuid:
70             self.seriesuid = generator.Generate()
71     def GetSponsorInitials(self):
72         return "dummy^foobar"
73     def GenerateStudyId(self):
74         return self.studyuid
75     def GenerateSeriesId(self):
76         return self.seriesuid
77     #def GenerateMSOPId(self):
78     def GenerateMSOPId(self):
79         generator = gdcm.UIDGenerator()
80         return generator.Generate()
81     def GetSiteId(self):
82         return "MySiteId"
83     def GetSiteName(self):
84         return "MySiteName"
85     def GetSponsorId(self):
86         return "MySponsorId"
87     def GetTPId(self):
88         return "MyTP"
89
90 if __name__ == "__main__":
91     import sys
92     gdcm.FileMetaInformation.SetSourceApplicationEntityTitle
93     ( "DumbAnonymizer" )
94     gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
95
96     r = gdcm.Reader()
97     filename = sys.argv[1]
98     r.SetFileName( filename )
99     if not r.Read(): sys.exit(1)
100
101     obj = MyAnon()
102
103     w = gdcm.Writer()
104     ano = gdcm.Anonymizer()
105     ano.SetFile( r.GetFile() )
106     ano.RemoveGroupLength()
107     for tag,rule in tag_rules.items():
108         if rule[0] == 'Value':
109             print tag,rule
110             ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
111         elif rule[0] == 'Method':
112             print tag,rule
113             # result = locals()[rule[1]]()
114             methodname = rule[1]
115             if hasattr(obj, methodname):
116                 _member = getattr(obj, methodname)
117                 result = _member()
118                 ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
119             else:
120                 print "Problem with: ", methodname
121
122     outfilename = sys.argv[2]
123     w.SetFileName( outfilename )
124     w.SetFile( ano.GetFile() )
125     if not w.Write(): sys.exit(1)

```

## 12.39 DumpADAC.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 goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.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, NULL }
};

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 << " ";
    assert( strcmp( magic, "adac01" ) == 0 );
    int c = is.get();
    assert( c == 0 ); (void)c;
    c = is.get();
    assert( c == 'X' );

    uint16_t v;
    v = readint16(is);
    // std::cout << v << std::endl;
    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;
            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 )
        {
            assert( diff == 4 );
            uint32_t val = readint32(is);
            os << " " << std::dec << val << " ";
        }
        else if( e.v2 == 0x300 )
        {
            assert( diff == 4 );
            float val = readfloat32(is);
            os << " " << std::dec << val << " ";
        }
        else
        {
            assert( 0 );
        }
        os << std::endl;
    }
}

```

```

    }
    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();

    // (0019,1061) UN (OB) 61\64\61\63\30          # 2048,1 Ver200 ADAC Pegasys Headers
    const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
    const gdcm::DataElement& ver200adacpegasysheaders = ds.
        GetDataElement( tver200adacpegasysheaders );
    if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
    const gdcm::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;

    gdcm::Element<gdcm::VR::IS,gdcm::VM::VM2> el;
    const gdcm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
    if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
    const gdcm::DataElement& ver200adacheaderimagesize = ds.
        GetDataElement( tver200adacheaderimagesize );
    el.SetFromDataElement( ver200adacheaderimagesize );
    if( el.GetValue(0) != 2048 ) return 1;

    std::iostream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );
    if( !b ) return 1;

    return 0;
}

```

## 12.40 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;
    }
}

```

## 12.41 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 (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:
* Triplett, William T for bringing to your attention on this ExamCard stuff
*/
#include "gdcmmReader.h"
#include "gdcmmDataSet.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmBase64.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 )
{
    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 = NULL;
    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;
        }
    }
    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) );
            assert( bla == 0x2 || bla == 0x3 );
            nstrings = 1;
            numparams = 1;
        } else {
            // indirect
            is.read( (char*)&nints, sizeof(nints));
            is.read( (char*)&v3, sizeof(v3));
            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));
            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));
assert( bla < sizeof(name0) );
is.read( name0, bla );
size_t l = strlen(name0);
assert( l == bla );
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];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla );
    size_t l = strlen(name0);
    assert( l == bla );
    memcpy( this->name, name0, bla );
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0x1 );
    is.read( (char*)&bla, sizeof(bla) );
    char value[32];
    memset(value,0,sizeof(value));
    assert( bla < sizeof(value) );
    is.read( value, bla );
    is.read( (char*)&bla, sizeof(bla) );
    assert( bla == 0 ); // trailing stuff ?
    is.read( (char*)&bla, sizeof(bla) );
    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 == 122 )
        is.read( (char*)&bla, 2 );
    else
        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:
    assert( strlen( name ) <= 32 );
    is.read( (char*)&boolean,1);
    assert( boolean == 0 || boolean == 1 || boolean == 0x69 ); // some kind of bool, or digital trash ?
    is.read( (char*)&type, sizeof( type ) );
    assert( gettypenamefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) ); // number of elements
    is.read( (char*)&v4.val, sizeof( v4.val ) );
    assert( v4.val == 0 ); // always 0 ? sometimes not...
    const uint32_t cur = (uint32_t)is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    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( gdcm::DataSet & ds )
{
    /*
    TODO:
    Looks like the real length of the blob is stored here:
(2005,1132) SQ                                     # u/1,1 ?
    (fffe,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 ?
    (fffe,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 gdcm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt0 ) ) return false;
    const gdcm::DataElement &de0 = ds.GetDataElement( pt0 );
    if( de0.IsEmpty() ) return false;
    const gdcm::ByteValue * bv0 = de0.GetByteValue();
    std::string s0( bv0->GetPointer() , bv0->GetLength() );

    // (2005,1139) LO [IEEE_PDF]                      # 8,1 Protocol Data Type
    const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt1 ) ) return false;
    const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

    // (2005,1143) SL 53                               # 4,1 Protocol Data Block Length
    (non-padded)
    const gdcm::PrivateTag pt2(0x2005,0x43,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt2 ) ) return false;
    const gdcm::DataElement &de2 = ds.GetDataElement( pt2 );

    // (2005,1147) CS [Y ]                             # 2,1 Protocol Data Boolean
    const gdcm::PrivateTag pt3(0x2005,0x47,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt3 ) ) return false;
    const gdcm::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 gdcm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt ) ) return false;
    const gdcm::DataElement &de = ds.GetDataElement( pt );
    if( de.IsEmpty() ) return false;
    const gdcm::ByteValue * bv = de.GetByteValue();

    if( s0 == "ExamCardBlob" )
    {
        assert( de1.IsEmpty() );

        std::string fn = gdcm::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 = gdc::Base64::GetDecodeLength(b64.c_str(), b64.size() );

std::string decoded;
decoded.resize( dlen );
gdc::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( del.IsEmpty() ) return false;
    const gdc::ByteValue * bv1 = del.GetByteValue();
    gdc::Element<gdc::VR::SL,gdc::VM::VM1> dlen = {{01}};
    dlen.SetFromDataElement( de2 );
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
    {
        std::istream is;
        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 );
        //assert( is.peek() && is.eof() );
    }
#ifdef 1
    static int c = 0;
    std::string fn0 = gdc::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

```

```

        {
            assert(0);
        }
        params.push_back( p );
    }
} else {
    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 = gdc::LOComp::Trim( s0.c_str() ); // remove trailing space
bool b1 = isvalidpdfstring( fn.c_str() );
assert( b1 ); (void)b1;
fn += ".csv";
//fn += ".xml";
std::ofstream csv( fn.c_str() );

// let's do some bookeeping:
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:
            ;
    }
}

#if 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

#endif
assert( h.getnints() >= nints );
assert( h.getnfloats() >= nfloats );
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 " )
{
    #if 0
    std::cerr << "ASCII is not handled" << std::endl;
    std::string fn = gdc::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 = gdc::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();
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( sl == "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();

#ifdef 0
    int array[ 128 ];
    memcpy( array, bv->GetPointer(), 512 );
    for( int i = 0; i < 14; ++i )
    {
        std::cout << array[i] << std::endl;
    }
#endif

    ret = true;
}
// else -> ret == false
assert( ret );

return ret;
}

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();
    /*
(2005,1132) SQ # u/1,1 ?
(fffe,e000) na (Item with undefined length)
(2005,0011) LO [Philips MR Imaging DD 002 ] # 26,1 Private Creator
(2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS] # 20,1 ?
(2005,1138) PN (LO) (no value) # 0,1 ?
(2005,1139) PN (LO) [IEEE_PDF] # 8,1 ?
(2005,1140) PN (LO) (no value) # 0,1 ?
(2005,1141) PN (LO) (no value) # 0,1 ?
(2005,1143) SL 3103 # 4,1 ?
(2005,1144) OW
66\05\00\00\3b\01\00\00\4a\0a\00\00\0e\00\00\00\7a\0a\00\00\95\01\00\00\08\00\00\00\1b\00\00\00\43\47\45\4e\5f\75\73\65\72\
# 3104,1 ?
(2005,1147) CS [Y ] # 2,1 ?
(fffe,e00d)
*/
    const gdcmm::PrivateTag pt(0x2005,0x32,"Philips MR Imaging DD 002");

```

```

if( !ds.FindDataElement( pt ) ) return 1;
const gdcm::DataElement &de = ds.GetDataElement( pt );
if( de.IsEmpty() ) return 1;

gdcm::SequenceOfItems *sqi = de.GetValueAsSQ();
if ( !sqi ) return 1;
gdcm::SequenceOfItems::SizeType s = sqi->
    GetNumberOfItems();
for( gdcm::SequenceOfItems::SizeType i = 1; i <= s; ++i )
{
    gdcm::Item &item = sqi->GetItem(i);

    gdcm::DataSet &nestedds = item.GetNestedDataSet();

    if( !ProcessNested( nestedds ) ) {
        std::cerr << "Error processing Item #" << i << std::endl;
    }
}

return 0;
}

```

## 12.42 DumpGEMSMovieGroup.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 <string>

#include <map>

bool PrintNameValueMapping( gdcm::SequenceOfItems *sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    using namespace gdcm;
    // prepare names mapping:
    typedef VRTToType<VR::UL::Type> UL;
    std::map< UL, std::string > names;
    assert( sqi_names );
    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 ) )
        {
            assert( 0 );
            return false;
        }
        const DataElement & index = ds.GetDataElement( tindex );
        const DataElement & name = ds.GetDataElement( tname );
    }
}

```



```

    if( index.IsEmpty() || name.IsEmpty() )
    {
        assert( 0 );
        return false;
    }
    gdcmm::Element<VR::UL, VM::VM1> el1;
    el1.SetFromDataElement( index );

    gdcmm::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();
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 ) )
    {
        assert( 0 );
        return false;
    }
    const DataElement & index2 = ds.GetDataElement( tindex2 );
    if( index2.IsEmpty() )
    {
        assert( 0 );
        return false;
    }
    gdcmm::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
    #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 tvaluefdl(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
        PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001"); // FD ??
    #if 1
        std::cout << " ) = ";
    #endif
    #endif
        if( ds.FindDataElement( tvalueint ) )
        {
            const DataElement & value = ds.GetDataElement( tvalueint );
            gdcmm::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 );
            gdcmm::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 );
            gdcmm::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( tvalues1 );
        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 );
    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 );
    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 );
    // 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 );
    // 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 );
    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 );
    assert( el2.GetLength() == 4 );
    std::cout << std::endl;
}
else
{
    std::cout << "(no value)" << std::endl;
    // std::cout << ds << std::endl;
    assert( ds.Size() == 2 );
}
}

```

```

    return true;
}

bool PrintNameValueMapping2( gdcmm::PrivateTag const & privtag, const
    gdcmm::DataSet & ds ,
    gdcmm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag ) ) return false;
    const gdcmm::DataElement& seq_values = ds.GetDataElement( privtag );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = seq_values.
        GetValueAsSQ();

    return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcmm::PrivateTag const & privtag1,
    gdcmm::PrivateTag const & privtag2, const gdcmm::DataSet & ds ,
    gdcmm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag1 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcmm::DataElement& values10name = ds.GetDataElement( privtag1 );
    gdcmm::Element<gdcmm::VR::LO,gdcmm::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( gdcmm::DataSet const & ds10, gdcmm::SequenceOfItems *sqi_dict
    , std::string const & indent )
{
    const gdcmm::PrivateTag tseq_values73(0x7fel,0x73,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values73 ) )
    {
        std::cout << indent << "No group 73" << std::endl;
        return false;
    }
    const gdcmm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73
    );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values73 =
        seq_values73.GetValueAsSQ();

    size_t ni3 = sqi_values73->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcmm::Item &item_73 = sqi_values73->GetItem(i3);
        gdcmm::DataSet &ds73 = item_73.GetNestedDataSet();
        assert( ds73.Size() == 3 );

        const gdcmm::PrivateTag tseq_values74name(0x7fel,0x74,"GEMS_Ultrasound_MovieGroup_001");
        const gdcmm::PrivateTag tseq_values75(0x7fel,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(0x7fel,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();
    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();
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");
assert( ds36.FindDataElement( timagedata ) );
gdcmm::DataElement const & imagedata = ds36.GetDataElement( timagedata );

const gdcmm::ByteValue * bv = imagedata.GetByteValue();
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();
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 ) )
{
assert( 0 );
return false;
}
const gdcmm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values10 =
seq_values10.GetValueAssQ();

size_t nil = sqi_values10->GetNumberOfItems();
// assert( nil == 1 );
for( size_t i1 = 1; i1 <= nil; ++i1 )
{
gdcmm::Item &item_10 = sqi_values10->GetItem(i1);
gdcmm::DataSet &ds10 = item_10.GetNestedDataSet();
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 gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
if( !ds10.FindDataElement( tseq_values20 ) )
{
    assert( 0 );
    return false;
}
const gdcm::DataElement& seq_values20 = ds10.GetDataElement(
    tseq_values20 );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 =
    seq_values20.GetValueAsSQ();

size_t ni2 = sqi_values20->GetNumberOfItems();
//assert( ni == 1 );
for( size_t i2 = 1; i2 <= ni2; ++i2 )
{
    gdcm::Item &item_20 = sqi_values20->GetItem(i2);
    gdcm::DataSet &ds20 = item_20.GetNestedDataSet();
    size_t count = ds20.Size(); (void)count;
    assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
    // (7fe1,0010)
    // (7fe1,1024)
    // (7fe1,1026)
    // (7fe1,1036)
    // (7fe1,103a)
    // (7fe1,1083) (*)

    const gdcm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001"
    );
    const gdcm::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 gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::File &file = reader.GetFile();
    gdcm::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();
    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;
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;
}

```

## 12.43 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"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct element
{
    std::istream & read( std::istream & is );

```

```

},

std::istream & element::read( std::istream & is )
{
    static const uint32_t ref = 0xe000fffe;
    std::ostream &os = std::cout;
    if ( is.eof() )
    {
        return is;
    }
    uint32_t magic;
    if ( !is.read( (char*)&magic, sizeof(magic) ) )
    {
        return is;
    }
    //os << magic << std::endl;
    assert( magic == ref ); (void)ref;

    uint32_t l;
    is.read( (char*)&l, sizeof(l) );
    //os << l << std::endl;

    char str[17];
    str[16] = 0;
    is.read( str, 16 );
    os << str << " (" << l << ")" << std::endl;
    std::vector<char> bytes;
    bytes.resize( 1 - 16 );
    if ( bytes.size() )
    {
        is.read( &bytes[0], 1 - 16 );
    }
    //os << "pos:" << is.tellg() << std::endl;

    if ( strcmp(str, "TUSREMEASUREMENT" ) == 0 )
    {
        const char *p = &bytes[0];
        uint32_t val;
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
        memcpy( (char*)&f, p, sizeof(f) );
        os << " " << f << std::endl;
        p += sizeof(f);
    #else
        memcpy( (char*)&val, p, sizeof(val) );
        os << " " << val << std::endl;
        p += sizeof(val);
    #endif

    memcpy( (char*)&val, p, sizeof(val) );
    os << " " << val << std::endl;
    p += sizeof(val);
    char str2[17];
    memcpy( str2, p, 16 );
    str2[16] = 0;
    os << " " << str2 << std::endl;
}

#if 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 reflen )
{
    // TUSNONIMAGESTAM (5176)
    // TUSREMEASUREMEN (1352)
    // TUSBSINGLELAYOU (16)
    // TUSCLIPPARAMETE (104)

    element el;
    while( el.read( is ) )
    {
    }
    //size_t pos = is.tellg();
    //assert( pos == reflen );
    (void)reflen;

    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();

    const gdcm::PrivateTag timageheaderinfo(0x0029,0x10,"TOSHIBA MDW HEADER");
    if( !ds.FindDataElement( timageheaderinfo ) ) return 1;
    const gdcm::DataElement& imageheaderinfo = ds.GetDataElement(
        timageheaderinfo );
    if ( imageheaderinfo.IsEmpty() ) return 1;
    const gdcm::ByteValue * bv = imageheaderinfo.GetByteValue();

    std::ostringstream 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 = 0.00416666668839752674; // 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;
}

```

## 12.44 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 *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::streamoff totalsize;

```

```

is.read( (char*)&totalsize, sizeof( totalsize ));
assert( totalsize == len );

uint32_t nframes;
is.read( (char*)&nframes, sizeof( nframes ));
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 );

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 ));

    assert( header == crchheaders[r] );
    assert( header.val1[0] == 2000 );
    assert( header.val1[1] == 3 );
    assert( header.val2[0] == 1 );
    assert( header.val2[1] == 1280 );

    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)&outbuf[0];
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (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);
    assert( ret == Z_OK ); (void)ret;
    assert( destLen >= (uLongf)size[0] * size[1] ); // 16bytes padding ?
    assert( header.imgsize == (uint32_t)size[0] * size[1] );
    //os.write( &outbuf[0], outbuf.size() );
    os.write( &outbuf[0], size[0] * size[1] );

    // skip data:
    is.seekg( sourceLen, std::ios::cur );
}
os.close();
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 > crchheaders;
    crchheaders.reserve( nslices );
    {
        std::istringstream 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
    crchheaders.push_back( header );
}

std::stringstream is;
is.str( std::string( buf, (size_t)len ) );

std::streampos totalsize;
is.read( (char*)&totalsize, sizeof( totalsize ));
assert( totalsize == len );

uint32_t nframes;
is.read( (char*)&nframes, sizeof( nframes ));
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 << crchheaders[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[0];

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
assert( header == crchheaders[r] );

is.read( buffer, buf_size - 16 );
os.write( buffer, header.imgsize );
}
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();
    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) IO 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();
        assert( sqi2->GetNumberOfItems() >= 1 );
    }
}

```

```

// FIXME: what if not in first Item ?
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();
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();
assert( zallocref >= 0 );
// (200d,3021) IS 2 0
const PrivateTag tzzero(0x200d,0x3021,"Philips US Imaging DD 033");
if( !ds3.FindDataElement( tzzero ) ) return 1;
const DataElement& zero = ds3.GetDataElement( tzzero );
Element<VR::IS,VM::VM1> elzero;
elzero.SetFromDataElement( zero );
const int zerocref = elzero.GetValue();
assert( zerocref == 0 ); (void)zerocref;

// (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();
assert( is_valid(outfilename) );
if( bv2 )
{
    assert( bv3 );
    assert( zallocref > 0 );
    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( outfile, 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;
}

```

## 12.45 DumpSiemensBase64.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.

=====*/
/*
 * https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ21LP8EcM/WzjFrtjnAgAJ
 */
#include "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlPrinter.h"
#include "gdcmlDictPrinter.h"
#include "gdcmlCSAHeader.h"
#include "gdcmlBase64.h"
#include "gdcmlExplicitDataElement.h"
#include "gdcmlSwapper.h"
#include "gdcmlPrinter.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];
    gdcml::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdcml::DataSet& ds = reader.GetFile().GetDataSet();

    gdcml::CSAHeader csa;
    const gdcml::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 gdcml::CSAElement &el = csa.GetCSAElementByName( name );
        const gdcml::ByteValue* bv = el.GetByteValue();
        std::string str( bv->GetPointer(), bv->GetLength() );
        str.erase( str.begin(), str.end(), '\n' );
        size_t dl = gdcml::Base64::GetDecodeLength( str.c_str(), str.size() );
        std::vector<char> buf;
        buf.resize( dl );
    }
}

```

```

size_t dl2 = gdc::Base64::Decode( &buf[0], buf.size(), str.c_str(), str.size() );
(void)dl2;
std::stringstream ss;
ss.str( std::string(&buf[0], buf.size()) );
gdc::File file;
gdc::DataSet &ds2 = file.GetDataSet();
gdc::DataElement xde;
try
{
    while( xde.Read<gdc::ExplicitDataElement,
            gdc::SwapperNoOp>( ss ) )
    {
        ds2.Insert( xde );
    }
    assert( ss.eof() );
}
catch( std::exception & )
{
    return 1;
}
gdc::Printer p;
p.SetFile( file );
p.Print( std::cout );
}

return 0;
}

```

## 12.46 DumpToshibaDTI.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.
=====*/
/*
 * https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu8lVvAMJ
 */
#include "gdcReader.h"
#include "gdcPrivateTag.h"
#include "gdcPrinter.h"
#include "gdcDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

bool DumpToshibaDTI( const char * input, size_t len )
{
    if( len % 2 ) return false;

    std::vector<char> copy( input, input + len );
    std::reverse( copy.begin(), copy.end() );

    std::istringstream is;
    std::string dup( &copy[0], copy.size() );
    is.str( dup );

    gdc::Reader reader;
    reader.SetStream( is );
    if( !reader.Read() )
        return false;

    //std::cout << reader.GetFile().GetDataSet() << std::endl;

```

```

//gdc::DictPrinter p;
gdc::Printer p;
p.SetFile( reader.GetFile() );
p.SetColor( true );
p.Print( std::cout );

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    const char *filename = argv[1];
    gdc::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Failed to read: " << filename << std::endl;
        return 1;
    }
    const gdc::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 gdc::PrivateTag tpmf(0x0029,0x1,"PMTF INFORMATION DATA");
    if( !ds.FindDataElement( tpmf ) ) return 1;
    const gdc::DataElement& pmtf = ds.GetDataElement( tpmf );
    if( pmtf.IsEmpty() ) return 1;
    gdc::SmartPointer<gdc::SequenceOfItems> seq = pmtf.
        GetValueAsSQ();
    if( !seq || !seq->GetNumberOfItems() ) return 1;

    size_t n = seq->GetNumberOfItems();
    for( size_t i = 1; i <= n; ++i )
    {
        gdc::Item &item = seq->GetItem(i);
        gdc::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 gdc::PrivateTag tseq(0x0029,0x90,"PMTF INFORMATION DATA");

        if( subds.FindDataElement( tseq ) )
        {
            const gdc::DataElement &de = subds.GetDataElement( tseq );
            const gdc::ByteValue *bv = de.GetByteValue();
            if( !bv ) return 1;

            bool b = DumpToshibaDTI( bv->GetPointer(), bv->GetLength() );
            if( !b ) return 1;
        }
    }

    return 0;
}

```

## 12.47 DumpToSQLITE3.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.

=====*/
/*
* 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(0);

    gdcm::Trace::SetDebug( false );
    gdcm::Trace::SetWarning( false );
    const char *inputdirectory = argv[1];

    gdcm::Directory d;
    unsigned int nfiles = d.Load( inputdirectory, true);

    gdcm::Scanner s;
    using gdcm::Tag;
    s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
    s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

    bool b0 = s.Scan( d.GetFilesNames() );
    if( !b0 ) return 1;
    time_t time_scanner = time(0);

    std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;

    // MappingType const &mappings = s.GetMappings();

    sqlite3* db;
    sqlite3_open("./dicom.db", &db);

    if(db == 0)
    {
        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, 0, 0, &errmsg);

    if(ret != SQLITE_OK)
    {
        printf("Error in statement: %s [%s].\n", sql_stmt, errmsg);
        return 1;
    }
    using gdcm::Directory;
    using gdcm::Scanner;
    const Directory::FileNamesType& files = d.GetFilesNames();
    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,
        0 // 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(0);

    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;
}

```

## 12.48 DuplicatePCDE.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 "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:  
 gdcConformanceTests/CT1\_J2KI\_DuplicatePCDE.dcm

Original thread can be found at:

[http://groups.google.com/group/comp.protocols.dicom/browse\\_thread/thread/82f28c4db28963af](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
*/
    gdcm::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 ...)
    gdcm::DataSet dup;
    gdcm::Tag new_private(0x0009,0x0);
    while (start.GetGroup() == 0x9 )
    {
        const gdcm::DataElement& de = ds.FindNextDataElement(start);
        const gdcm::Tag &t = de.GetTag();
        if( t.IsPrivateCreator() )
        {
            std::cout << t << std::endl;
            // Ok let's duplicate into the next available attribute:
            gdcm::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;
            gdcm::DataElement duplicate = de;
            duplicate.GetTag().SetPrivateCreator( new_private );
            if( const gdcm::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:
                gdcm::ByteValue *dupbv = new gdcm::ByteValue( bv->GetPointer(),
                    bv->GetLength() );
                // Let's recognize the duplicated ASCII-type elements:
                if( duplicate.GetVR() & gdcm::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) );
        }
    }

    gdcm::DataSet::ConstIterator it = dup.Begin();
    for( ; it != dup.End(); ++it )
    {
        ds.Insert( *it );
    }
}

```

```

    }

    gdcm::Writer w;
    w.SetFile( file );
    w.SetFileName( outfilename );
    if (!w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.49 ELSCINT1WaveToText.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 "gdcmPrivateTag.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 (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:
 * 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 )
{
    short * buffer = (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' << " " << '\\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+74] << '\t' << " "          << '\t' << " "          << '\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' << " "
    << '\t' << buffer[i+74] << '\t' << " "          << '\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' << " "
    << '\t' << buffer[i+74] << '\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+74] << '\t' << " "          << '\t' << buffer[i+74]          << '\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' << " "
    << '\t' << buffer[i+74] << '\t' << buffer[i+74]          << '\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' <<
buffer[i+74] << '\t' << " "          << '\t' << " "          << '\t' << " "          << '\t' << " "
        << std::endl;
    if (buffer[i+75] == 256)
        text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0          << '\t' << " "
        << '\t' << buffer[i+74] << '\t' << " "          << '\t' << " "          << '\t' << " "
        << std::endl;
    if (buffer[i+75] == -32768)
        text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1          << '\t' << " "
        << '\t' << buffer[i+74] << '\t' << " "          << '\t' << " "          << '\t' << " "
        << std::endl;
    if (buffer[i+75] == -16384)
        text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1          << '\t' <<
buffer[i+74] << '\t' << " "          << '\t' << buffer[i+74]          << '\t' << " "          << '\t' << " "
        << std::endl;
    if (buffer[i+75] == -32512)
        text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1          << '\t' << " "
        << '\t' << buffer[i+74] << '\t' << buffer[i+74]          << '\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();
    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;
}

```

## 12.50 EmptyMask.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 "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;
}

```

## 12.51 EncapsulateFileInRawData.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 "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( outfile );

    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: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

```

## 12.52 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 MIMETYPEOfEncapsulatedDocument
 * ...
 *
 * 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.FindDataElement( 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;
    }
}

```

## 12.53 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();
    gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::DataElement &EncryptedAttributesSequence = ds.
        GetDataElement( gdcm::Tag( 0x0400,0x0500 ) );

    gdcm::SequenceOfItems *sqi = EncryptedAttributesSequence.
        GetValueAssQ();

    if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

    gdcm::Item &item = sqi->GetItem(1);

    gdcm::DataSet &nestedds = item.GetNestedDataSet();

    if( ! nestedds.FindDataElement( gdcm::Tag( 0x0400,0x0520 ) ) ) return 1;

    const gdcm::DataElement &EncryptedContent = nestedds.
        GetDataElement( gdcm::Tag( 0x0400,0x0520 ) );

    const gdcm::ByteValue *bv = EncryptedContent.GetByteValue();

    std::ofstream of( outfile, std::ios::binary );
    of.write( bv->GetPointer(), bv->GetLength() );
    of.close();

    return 0;
}

```

## 12.54 ExtractIconFromFile.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 either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcm::IconImage& icon)
{
    gdcm::PNMCodec pnm;
    pnm.SetDimensions( icon.GetDimensions() );
    pnm.SetPixelFormat( icon.GetPixelFormat() );
    pnm.SetPhotometricInterpretation( icon.
        GetPhotometricInterpretation() );
    pnm.SetLUT( icon.GetLUT() );
    const gdcm::DataElement& in = icon.GetDataElement();
    bool b = pnm.Write( filename, in );
    assert( b );
    return b;
}

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 (or not image): " << filename << std::endl;
        return 1;
    }

    gdcm::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();
            assert( bv );
            std::ofstream out( "icon.jpg", std::ios::binary );
            out.write( bv->GetPointer(), bv->GetLength() );
            out.close();
        }
    }
    else

```

```

{
    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;
}

```

## 12.55 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;
}

```

## 12.56 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");
            }
        }
    }
}

```

## 12.57 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/rle16l00.dcm
* $ md5sum /tmp/frame_rgb.raw
* 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
* $ gdcming --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
* $ gdcmvviewer 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;
}

```

## 12.58 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.decode_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 */
    // gdcData/ELSCINT1_JP2vsJ2K.dcm
    // gdcData/MAROTECH_CT_JP2Lossy.dcm
    //gdcWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
    parameters.decode_format = 1; //JP2_CFMT;
    //assert(parameters.decode_format == JP2_CFMT);
}
else
{
    /* JPEG-2000 codestream */
    //parameters.decode_format = J2K_CFMT;
    //assert(parameters.decode_format == J2K_CFMT);
    assert( 0 );
}
parameters.cod_format = 11; // PGX_DFMT;
//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);
    //gdcErrorMacro( "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;
        }
    }

    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 del( gdcmm::Tag(0x8,0x16) );
    del.SetVR( gdcmm::VR::UI );
    gdcmm::MediaStorage ms( gdcmm::MediaStorage::CTImageStorage
        );
    del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
    ds.Insert( del );

    const char mystr[] = "MONOCHROME2 ";
    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 = {image->comps[0].w};
    //row.SetValue(512);
    ds.Insert( row.GetAsDataElement() );
    // w.SetCheckFileMetaInformation( true );
    gdcmm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
    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 = {image->numcomps};
    ds.Insert( at1.GetAsDataElement() );

    gdcmm::Attribute<0x0028,0x0101> at2 = {8};
    ds.Insert( at2.GetAsDataElement() );

    gdcmm::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)-1)
    {
        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> ell;
    ell.SetValue(col,0);
    ell.SetValue(row,1);
    gdcm::DataElement brr = ell.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( gdcm::VR::UI );
const char *ul = uid1.Generate();
dea.SetByteValue( ul, strlen(ul) );
ds1.Insert( dea );

gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
deb.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms1(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()) );
ds1.Insert( deb );

const char mystr1[] = "MONOCHROME2 ";
gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
dec.SetVR( gdcm::VR::CS );
dec.SetByteValue(mystr, strlen(mystr));
ds1.Insert( dec );

gdcm::Attribute<0x0028,0x0010> row1 = {image->y1};
//row.SetValue(512);
ds1.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col1 = {image->x1};
ds1.Insert( col1.GetAsDataElement() );
gdcm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> ata = {8};
ds1.Insert( ata.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> atb = {image->numcomps};

```

```

    ds1.Insert( atb.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0101> atc = {8};
    ds1.Insert( atc.GetAsDataElement() );

    gdcm::Attribute<0x0028,0x0102> atd = {7};
    ds1.Insert( atd.GetAsDataElement() );

    theStreamWriter.SetFile(file1);

    gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
    des.SetVR(gdcm::VR::SQ);
    //des.SetVR(gdcm::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<<"\nable to read";

// Important to write here
std::vector<unsigned int> extent = gdcm::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;
    gdcm::StreamImageWriter theStreamWriter;

    std::ofstream of;
    of.open( outfile, 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));
    assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    assert( of );

    return 0;
}

```

## 12.59 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;
}
}

```

## 12.60 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://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 "gdcmReader.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"

int main(int, char *[])
{
    char * buffer = new char[ 256 * 256 *3 ];
    // *p = (uint8_t*)buffer;
    char * p = buffer;

    gdcm::Trace::DebugOn();
    gdcm::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;
            }
    }

    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[] = "RGB";
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 = {256};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcm::Attribute<0x0028,0x0011> col = {256};
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 = {3}; //bits per pixel
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0006> at4 = {0};
ds.Insert( at4.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0103> at5 = {0};
ds.Insert( at5.GetAsDataElement() );

//de.SetTag(gdcm::Tag(0x7fe0,0x0010));
//ds.Insert(de);

gdcm::StreamImageWriter theStreamWriter;
gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new
    gdcm::SequenceOfItems();
sq->SetLengthToUndefined();

uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;

gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
el2.SetValue(1);
gdcm::DataElement rfn = el2.GetAsDataElement(); //rfn --->
    reference frame number
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> ell;
ell.SetValue(col1,0);
ell.SetValue(row1,1);
gdcm::DataElement brr = ell.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 theBufferSize = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize];
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));
assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize);
of.flush();
assert( of );

return 0;
}

```

## 12.61 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 gdcm;

public class FileAnonymize
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];

        gdcm.FileAnonymizer fa = new gdcm.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;
    }
}

```

## 12.62 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" );
    }
}

```

## 12.63 FileChangeTS.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 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 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 AssembledDICOMAndRaw(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;
}
}

```

## 12.64 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 AssembledDICOMAndRaw(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;
}
}

```

## 12.65 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;
    }
}
```

## 12.66 FindAllPatientName.py

```
1
14 """
15 This example shows how one can use the gdcm.CompositeNetworkFunctions class
16 for executing a C-FIND query
17 It will print the list of patient name found
18
19 Usage:
20
21 python FindAllPatientName.py
22
23 """
24
25 import gdcm
26
27 # Patient Name
28 tag = gdcm.Tag(0x10,0x10)
29 de = gdcm.DataElement(tag)
30
```

```

31 # Search all patient name where string match 'F*'
32 de.SetByteValue('F*',gdcml.VL(2))
33
34 ds = gdcml.DataSet()
35 ds.Insert(de)
36
37 cnf = gdcml.CompositeNetworkFunctions()
38 theQuery = cnf.ConstructQuery (gdcml.ePatientRootType,gdcml.ePatient,ds)
39
40 #print theQuery.ValidateQuery()
41
42 # prepare the variable for output
43 ret = gdcml.DataSetArrayType()
44
45 # Execute the C-FIND query
46 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
47
48 for i in range(0,ret.size()):
49     print "Patient #",i
50     print ret[i]

```

## 12.67 FixBrokenJ2K.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 "gdcmlReader.h"
#include "gdcmlWriter.h"
#include "gdcmlImageReader.h"
#include "gdcmlSequenceOfFragments.h"
#include "gdcmlFile.h"

// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvvua-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];
    gdcml::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {

```

```

    return 1;
}

gdcm::File &file = reader.GetFile();
const gdcm::DataElement &pixeldata0 = file.GetDataSet().
    GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
const gdcm::SequenceOfFragments *sqf = pixeldata0.
    GetSequenceOfFragments();
if( !sqf )
{
    return 1;
}
const gdcm::Fragment &frag0 = sqf->GetFragment(0);

const gdcm::ByteValue *bv = frag0.GetByteValue();
const char *ptr = bv->GetPointer();
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;
const 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:
        gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
        pixeldata.SetVR( gdcm::VR::OB );
        gdcm::SmartPointer<gdcm::SequenceOfFragments> sq = new
            gdcm::SequenceOfFragments;

        gdcm::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;
}

gdcm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
writer.CheckFileMetaInformationOff();
if( !writer.Write() )
{
    std::cerr << "Could not write" << std::endl;
}

// paranoid check:
gdcm::ImageReader ireader;
ireader.SetFileName( outfilename );
if( !ireader.Read() )
{
    std::cerr << "file written is still not valid, please report" << std::endl;
    return 1;
}

return 0;
}

```

## 12.68 FixCommaBug.py

```

1
14
15 """
16 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
17 "." as required by the DICOM standard
18 Issue is still current (IMHO) with gdcm 2.0.9
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 outname = sys.argv[2]
26
27 # read
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31     print "not valid"
32     sys.exit(1)
33
34 file = r.GetFile()
35 dataset = file.GetDataSet()
36
37 ano = gdcm.Anonymizer()
38 ano.SetFile( file )
39
40 tags = [
41     gdcm.Tag(0x0018,0x1164),
42     gdcm.Tag(0x0018,0x0088),
43     gdcm.Tag(0x0018,0x0050),
44     gdcm.Tag(0x0028,0x0030),
45 ]
46
47 for tag in tags:
48     print tag
49     if dataset.FindDataElement( tag ):
50         pixelpacing = dataset.GetDataElement( tag )
51         #print pixelpacing
52         bv = pixelpacing.GetByteValue()
53         str = bv.GetBuffer()
54         #print bv.GetLength()
55         #print len(str)
56         new_str = str.replace(",",".")
57         # Need to explicitly pass bv.GetLength() to remove any trailing garbage
58         ano.Replace( tag, new_str, bv.GetLength() )
59
60 #print dataset
61
62 w = gdcm.Writer()
63 w.SetFile( file )
64 w.SetFileName( outname )
65 if not w.Write():
66     print "Cannot write"
67     sys.exit(1)
68
69 # paranoid:
70 image_reader = gdcm.ImageReader()
71 image_reader.SetFileName( outname )
72 if not image_reader.Read():
73     print "there is still a comma"
74     sys.exit(1)
75
76 print "Sucess!"
77 sys.exit(0) # success

```

## 12.69 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 --jpegls' 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<BYTE> rgbbyteOutall;
    for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
    {

```

```

const gdcM::Fragment &frag = sf->GetFragment(i);
if( frag.IsEmpty() ) return 1;
const gdcM::ByteValue *bv = frag.GetByteValue();
if( !bv ) return 1;
unsigned long totalLen = bv->GetLength();

std::vector<char> vbuffer;
vbuffer.resize( totalLen );
char *buffer = &vbuffer[0];
bv->GetBuffer(buffer, totalLen);
const BYTE* pbyteCompressed0 = (const BYTE*)buffer;
while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
{
    totalLen--;
}

JlsParameters metadata;
if (JpegLsReadHeader(buffer, totalLen, &metadata) != OK)
{
    std::cerr << "Cant parse jpegls" << std::endl;
    return 1;
}

std::cout << metadata.width << std::endl;
std::cout << metadata.height << std::endl;
std::cout << metadata.bitspersample << std::endl;

gdcM::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 = NULL;
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 << "Cant 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[0];
    size_t cbyteCompressed = vbuffer.size(); // updated legnth

    JlsParameters params;
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params);

    std::vector<BYTE> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height * params.width * ((params.bitspersample + 7)
        / 8) * params.components);

    JLS_ERROR result =
        JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params );
    if (result != OK)
    {
        std::cerr << "Could not patch JAI-JPEGLS" << 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[0], (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;
}

```

## 12.70 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 );
#ifdef 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;
}

```



## 12.71 gdcmmorthoplanes.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 "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 "gdcmmSystem.h"
#include "gdcmmDirectory.h"
#include "gdcmmIPPSorter.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 && gdcm::System::FileIsDirectory( filename ) )
        {
            std::cout << "Loading directory: " << filename << std::endl;
            bool recursive = false;
            gdcm::Directory d;
            d.Load(filename, recursive);
            gdcm::Directory::FileNamesType const &files = d.
            GetFileNames();
            for( gdcm::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( gdcm::System::FileExists( filename ) )
                {
                    if( gdcm::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 )
{
    //gdcm::Trace::DebugOn();
    //gdcm::Trace::WarningOn();
    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;
    ippzspacing = s.GetZSpacing();

    const std::vector<std::string> & sorted = s.GetFilenames();
    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 *vl6 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
    vl6->SetInputConnection( reader->GetOutputPort() );
#else
    vl6->SetInput( reader->GetOutput() );
#endif
vl6->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
vl6->Update();

#if 0
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetInput( vl6->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(vl6->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
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();
#if (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();
#if (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 w1[2];
    planeWidgetZ->GetWindowLevel(w1);

    // Add a 2D image to test the GetReslice method
    //
    vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
    colorMap->PassAlphaToOutputOff();
    colorMap->SetActiveComponent(0);
    colorMap->SetOutputFormatToLuminance();
    #if (VTK_MAJOR_VERSION >= 6)
    colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
    #else
    colorMap->SetInput(planeWidgetZ->GetResliceOutput());
    #endif
    colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

    vtkImageActor* imageActor = vtkImageActor::New();
    imageActor->PickableOff();
    #if (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' postions
    //
    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();
vl6->Delete();
reader->Delete();

return 0;
}

```

## 12.72 gdcmlreslice.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 "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"

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 ); // cant 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;
}

```

## 12.73 gdcmrptionplan.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 "gdcmsReader.h"
#include "gdcmsAttribute.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];

    gdcms::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcms::MediaStorage ms;
    ms.SetFromFile( reader.GetFile() );
    if( ms != gdcms::MediaStorage::RTIonPlanStorage )

```

```

    {
        return 1;
    }

/*
(300a,03a2) SQ                                     # u/1,1 Ion Beam Sequence
(fffe,e000) na (Item with undefined length)
(0008,i040) 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
(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\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\
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
(fffe,e000)

*/
const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
gdcm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcm::DataElement &tbeamsq = ds.GetDataElement( tbeamsq );
//std::cout << tbeamsq << 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(1); // Item start at #1
//     const gdcm::DataSet& nestedds = item.GetNestedDataSet();
//     //std::cout << nestedds << std::endl;
//     gdcm::Tag tcompensatorsq(0x300a,0x02ea);
//     if( !nestedds.FindDataElement( tcompensatorsq ) )
//     {
//         return 1;
//     }
//     const gdcm::DataElement &tcompensatorsq = nestedds.
//         GetDataElement( tcompensatorsq );
//     //std::cout << tcompensatorsq << 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;
    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( (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);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::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\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7
    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 &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 &blockdata = 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] = 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;
}

```

## 12.74 gdcmrtpplan.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 <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmsReader.h"
#include "gdcmsAttribute.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
      (ffff,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
          (ffff,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 [3.679999\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\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\
              Data
                (300a,02e0) CS [ABSENT]                  # 6,1 Compensator Divergence
                (300a,02e1) CS [SOURCE_SIDE ]            # 12,1 Compensator Mounting Position
            (ffff,e00d)
              (ffff,e000) na (Item with undefined length)
              (ffff,e00d)
            (ffff,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 << tbeamsq << 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 &compensatorsq = nestedds.
    GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = compensatorsq
    .GetValueAssSQ();
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 &compensatorthicknessdata = nestedds2.
    GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1 Compensator Rows
gdcm::Attribute<0x300a,0x00e7> at1;
const gdcm::DataElement &compensatorrows = nestedds2.
    GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
// (300a,00e8) IS [37] # 2,1 Compensator Columns
gdcm::Attribute<0x300a,0x00e8> at2;
const gdcm::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
gdcm::Attribute<0x300a,0x00e9> at3;
const gdcm::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
gdcm::Attribute<0x300a,0x00ea> at4;
const gdcm::DataElement &compensatorposition = nestedds2.
    GetDataElement( at4.GetTag() );
at4.SetFromDataElement( compensatorposition );
std::cout << at4.GetValue(0) << std::endl;

vtkDoubleArray *d = vtkDoubleArray::New();
d->SetArray( (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();
#if (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

```

```

        (fffe,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\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\
        (fffe,e00d)
        (fffe,e000) na (Item with undefined length)
        (fffe,e00d)
        (fffe,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 &blockdata = 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] = 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->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;
}

```

## 12.75 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"

```

```

// 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;
}

```

## 12.76 gdcmttexture.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 "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"

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();
#if (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();
#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);

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 ); // cant get it to work
cube->GetAssembly()->SetUserTransform( transform ); // cant 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;
}

```

## 12.77 gdcmvolume.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 "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"

// 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;
}

```

## 12.78 GenAllVR.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 "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 gdcmm::Dict;
    Dict::ConstIterator beg = dict.Begin();
    Dict::ConstIterator end = dict.End();
    Dict::ConstIterator it;
    for( it = beg; it != end; ++it)
    {
        const gdcmm::Tag &t = it->first;
        const gdcmm::DictEntry &de = it->second;
        const gdcmm::VR &vr_de = de.GetVR();
        if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
        {
            return t;
        }
    }
    return gdcmm::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 gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();
    using gdcmm::VR;
    using gdcmm::Tag;

    gdcmm::Writer w;

    gdcmm::File &f = w.GetFile();
    gdcmm::DataSet &ds = f.GetDataSet();

    gdcmm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }

    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
        gdcmm::SequenceOfItems();
    sq->SetLengthToUndefined();

    // gdcmm::DummyValueGenerator dv;

    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcmm::DataElement owner( gdcmm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcmm::VR::LO );

    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    gdcmm::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.SetVLTToUndefined();

ds.Insert(owner);
ds.Insert(des);

// avoid INVALID = 0
for(int i = 1; i < 27; ++i)
{
    VR vr = (VR::VRType)(1 << i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
    {
        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(
    gdcm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.79 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://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 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.GetFilenames();

        gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
        gen.SetFilenames( filenames );
        gen.SetDescriptor( descriptor );
        if( !gen.Generate() )
        {
            return 1;
        }

        gdcm.FileMetaInformation.
            SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
        gdcm.Writer writer = new Writer();
        writer.SetFile( gen.GetFile() );
        writer.SetFileName( outfilename );
        if( !writer.Write() )
        {
            return 1;
        }

        return 0;
    }
}

```

## 12.80 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 <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::FilenameType 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::FilenameType 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";
        gdcm::Directory::FilenameType 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];
            sprintf(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 blank 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;
}

```

## 12.81 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;
}

```

## 12.82 GenFakelIdentifyFile.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 "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"

#include <cstdlib>
#include <cstring>

gdcm::DataElement CreateFakeElement(gdcm::Tag const &tag, bool toremove)
{

```

```

static const gdcm::Global &g = gdcm::Global::GetInstance();
static const gdcm::Dicts &dicts = g.GetDicts();
static const gdcm::Dict &pubdict = dicts.GetPublicDict();
static size_t countglobal = 0;
static std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
    ();
size_t count = countglobal % balcptags.size();

const gdcm::DictEntry &dictentry = pubdict.GetDictEntry(tag);

gdcm::DataElement de;
de.SetTag( tag );
using gdcm::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
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    assert(de.GetVR() == gdcm::VR::SQ );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new
        gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
    de.SetVLToUndefined();
    //ds.Insert(de);

    if( !toremove )
    {
        nds.Insert( CreateFakeElement( balcptags[count], true ) );
        countglobal++;
    }
    else
    {
        gdcm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no
            reason to be 'anonymized'...
        nds.Insert( at1.GetAsDataElement() );
        gdcm::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 gdcm::Tag;
using gdcm::VR;
const char *outfilename = argv[1];

std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
        ();

gdcm::Writer w;
gdcm::File &f = w.GetFile();
gdcm::DataSet &ds = f.GetDataSet();

// Add attribute that need to be anonymized:
std::vector<gdcm::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 gdcm::Global &g = gdcm::Global::GetInstance();
static const gdcm::Dicts &dicts = g.GetDicts();
static const gdcm::Dict &pubdict = dicts.GetPublicDict();

using gdcm::Dict;
Dict::ConstIterator dictit = pubdict.Begin();
for( ; dictit != pubdict.End(); ++dictit )
{
    const gdcm::Tag &dicttag = dictit->first;
    if( dicttag == Tag(0x6e65,0x6146) ) break;
    //const gdcm::DictEntry &dictentry = dictit->second;
    ds.Insert( CreateFakeElement( dicttag, false ) );
}
ds.Remove( gdcm::Tag(0x400,0x500) );
ds.Remove( gdcm::Tag(0x12,0x62) );
ds.Remove( gdcm::Tag(0x12,0x63) );

// 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 );
ds.Replace( 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.Replace( de ); // replace !

gdcm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
fmi.SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfile );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.83 GenFakelImage.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 "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
//#include "gdcmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcm.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
    gdcm::SmartPointer<gdcm::Image> im = new
        gdcm::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;
}

```

## 12.84 GenLongSeqs.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:
 *
 *
 */

```

```

* There is a flaw in the DICOM design were 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 nitems = 1000;
    nitems += std::numeric_limits<uint32_t>::max();
    for( unsigned int idx = 0; idx < 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( outfile );
    if( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 12.85 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; */
    //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;
}

```

## 12.86 GetArray.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/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 r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
    //byte[] rbuf = new byte[ r1 ];
    //uint r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
    //assert r1 == r12;

    //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;
}
}

```

## 12.87 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;
}

```

## 12.88 GetPortionCSAHeader.py

```

1
14
15 """
16 Usage:
17
18 python GetPortionCSAHeader.py input.dcm
19
20 Footnote:
21     SIEMENS is not publishing any information on the CSA header. So any info extracted
22     is at your own risk.
23 """
24
25 import sys
26 import gdcm
27
28 if __name__ == "__main__":
29
30     file = sys.argv[1]
31
32     r = gdcm.Reader()
33     r.SetFileName( file )
34     if not r.Read():
35         sys.exit(1)
36
37     ds = r.GetFile().GetDataSet()
38     csa_t1 = gdcm.CSAHeader()
39     csa_t2 = gdcm.CSAHeader()
40     #print csa
41     t1 = csa_t1.GetCSAImageHeaderInfoTag();
42     print t1
43     t2 = csa_t2.GetCSASeriesHeaderInfoTag();
44     print t2
45     # Let's do it for t1:
46     if ds.FindDataElement( t1 ):
47         csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
48         print csa_t1
49
50     # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
51     bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
52     print bvalues
53
54     diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive
55     !
56     print diffgraddir
57
58     # repeat for t2 if you like it:
59     if ds.FindDataElement( t2 ):
60         csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
61         # print csa_t2
62
63     gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
64     print gdt
65

```

```

65  bv = gdt.GetByteValue();
66  #print bv
67  str = bv.GetPointer()
68  print str.split("\\\\")

```

## 12.89 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;
}

```

## 12.90 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 <string>

#include <map>

/*
 * This example will extract the Movie from the private group of
 * GEMS_Ultrasound_MovieGroup_001 See Attribute
 * (7fel,60,GEMS_Ultrasound_MovieGroup_001)
 *
 * The output file will be stored in 'outvid.dcm' as
 * MultiframeGrayscaleByteSecondaryCaptureImageStorage
 */
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 );
    reader.Read();

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataElement();
    const PrivateTag tseq(0x7fel,0x1,"GEMS_Ultrasound_MovieGroup_001");

    if( !ds.FindDataElement( tseq ) ) return 1;
    const DataElement& seq = ds.GetDataElement( tseq );

    SmartPointer<SequenceOfItems> sqi = seq.
        GetValueAsSQ();
    assert( sqi->GetNumberOfItems() == 1 );
    Item &item = sqi->GetItem(1);
    DataSet &subds = item.GetNestedDataSet();

    const PrivateTag tseq1(0x7fel,0x10,"GEMS_Ultrasound_MovieGroup_001");

    if( !subds.FindDataElement( tseq1 ) ) return 1;
    const DataElement& seq1 = subds.GetDataElement( tseq1 );

    SmartPointer<SequenceOfItems> sqi2 = seq1.
        GetValueAsSQ();
    //int n = sqi2->GetNumberOfItems();
    int index = 1;
    Item &item2 = sqi2->GetItem(index);
    DataSet &subds2 = item2.GetNestedDataSet();

    const PrivateTag tseq2(0x7fel,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;
    assert( sqi3->GetNumberOfItems() >= 1 );
    Item &item3 = sqi3->GetItem(1);
    DataSet &subds3 = item3.GetNestedDataSet();

    const PrivateTag tseq6(0x7fel,0x26,"GEMS_Ultrasound_MovieGroup_001");
    if( !subds3.FindDataElement( tseq6 ) ) return 1;
    const DataElement& seq6 = subds3.GetDataElement( tseq6 );
    SmartPointer<SequenceOfItems> sqi6 = seq6.
        GetValueAsSQ();
    size_t ni6= sqi6->GetNumberOfItems();

```

```

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 ) );
        std::cout << "El= " << el.GetValue() << std::endl;
        dimx = el.GetValue(0);
        dimy = el.GetValue(1);
    }
}

const PrivateTag tseq3(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
if( !subds3.FindDataElement( tseq3 ) ) return 1;
const DataElement& seq3 = subds3.GetDataElement( tseq3 );

//      std::cout << seq3 << std::endl;

SmartPointer<SequenceOfItems> sqi4 = seq3.
    GetValueAsSQ();
size_t ni4= sqi4->GetNumberOfItems();
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 1;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL, VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return 1;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return 1;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );

    //      std::cout << seq4 << std::endl;
    //      std::cout << seq5 << std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
    #if 0
    {
        std::ofstream out( "/tmp/mo4", std::ios::binary );
        out.write( bv4->GetPointer(), bv4->GetLength());
        out.close();
    }
    #endif
    const ByteValue *bv5 = seq5.GetByteValue();
    #if 0
    {
        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[0], (uint32_t)imbuffer.size() );

gdcm::SmartPointer<gdcm::Image> im = new
    gdcm::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;
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( "outvid.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 12.91 headsq2dcm.py

```

1
14
15 """
16 Usage:
17 python headsq2dcm.py -D /path/to/VTKData
18 """
19
20 import vtk
21 import vtkgdcmm
22 from vtk.util.misc import vtkGetDataRoot
23 VTK_DATA_ROOT = vtkGetDataRoot()
24
25 reader = vtk.vtkVolume16Reader()
26 reader.SetDataDimensions(64, 64)
27 reader.SetDataByteOrderToLittleEndian()
28 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsquarter")
29 reader.SetImageRange(1, 93)
30 reader.SetDataSpacing(3.2, 3.2, 1.5)
31
32 cast = vtk.vtkImageCast()
33 cast.SetInput( reader.GetOutput() )
34 cast.SetOutputScalarTypeToUnsignedChar()
35
36 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
37 writer = vtkgdcmm.vtkGDCMImageWriter()
38 writer.SetFileName( "headsq.dcm" )
39 writer.SetInput( reader.GetOutput() )
40 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
41 #writer.SetInput( cast.GetOutput() )
42 writer.SetFileDimensionality( 3 )
43 writer.Write()

```

## 12.92 HelloActiviz.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 vtkgdcm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcm 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/gdcmData/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(vtkgdcm.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
        return imgout;
    }

    static vtkgdcm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdcm.vtkImageData imgout = new vtkgdcm.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

vtkgdcmm.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;
}
}

```

## 12.93 HelloActiviz2.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 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 gdcmmData/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;
}
}

```

## 12.94 HelloActiviz3.cs

```

/*=====
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.

=====*/
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.WriteLine(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;
    }
}

```

## 12.95 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.WriteLine(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;
}
}

```

## 12.96 HelloActiviz5.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;

// 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.WriteLine( "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.WriteLine( "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.WriteLine( "dataRoot: " + dataRoot + "\n" );
        System.Console.WriteLine( "filename being used is: " + filename + "\n" );

        vtkGDCMImageReader reader = vtkGDCMImageReader.

```

```

        New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);
        reader.SetFileNames(array);
        reader.Update();

        System.Console.WriteLine(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;
    }
}

```

## 12.97 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 );
}
}

```

## 12.98 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;
}

```

## 12.99 HelloVTKWorld.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 vtkgdcm;

/*
 * 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() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
            {
                System.Console.WriteLine( "Image is MONOCHROME2" ); //
            }

            // Just for fun, invert the direction cosines, output should reflect that:
            vtkMatrix4x4 dirsos = reader.GetDirectionCosines();
            dirsos.Invert();

            string outfilename = args[1];
            vtkGDCMImageWriter writer = vtkGDCMImageWriter.
                New();
            writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
            writer.SetDirectionCosines( dirsos );
            writer.SetShift( reader.GetShift() );
            writer.SetScale( reader.GetScale() );
            writer.SetImageFormat( reader.GetImageFormat() );
            writer.SetFileName( outfilename );
            writer.SetInputConnection( reader.GetOutputPort() );

```

```

        writer.Write();

        return 0;
    }
}

```

## 12.100 HelloVTKWorld.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' eventhough I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcml.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdcml.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:vtkgdcml.jar:gdcml.jar:. java HelloVTKWorld gdcmlData/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("vtkgdcmlJava");
        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() == vtkgdcml.vtkgdcml.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 );
}
}
```

## 12.101 HelloVTKWorld2.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 vtkgdcm;

/*
 * 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();

        vtkVolumel6Reader reader = vtkVolumel6Reader.New();
        reader.SetDataDimensions(64, 64);
        reader.SetDataByteOrderToLittleEndian();
        reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/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( "headsq.dcm" );
        writer.SetInputConnection( reader.GetOutputPort() );
        // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
        // writer.SetInputConnection( cast.GetOutputPort() );
    }
}
```

```

        writer.SetFileDimensionality( 3 );
        writer.Write();

        return 0;
    }
}

```

## 12.102 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:
    gdcm::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 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();

    // Construct a static(*) type for Image Comments :
    gdcm::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
    gdcm::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( outfile );
    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.
 */

```

## 12.103 HelloWorld.py

```

1
14
15 """
16 Hello World !
17 """
18
19 import gdcm
20 import sys
21
22 if __name__ == "__main__":
23
24     # verbosity:
25     #gdcm.Trace.DebugOn()
26     #gdcm.Trace.WarningOn()
27     #gdcm.Trace.ErrorOn()
28
29     # Get the filename from the command line
30     filename = sys.argv[1]
31
32     # Instantiate a gdcm.Reader
33     # This is the main class to handle any type of DICOM object
34     # You should check for gdcm.ImageReader for reading specifically DICOM Image file
35     r = gdcm.Reader()
36     r.SetFileName( filename )
37     # If the reader fails to read the file, we should stop !
38     if not r.Read():
39         print "Not a valid DICOM file"
40         sys.exit(1)
41
42     # Get the DICOM File structure
43     file = r.GetFile()
44
45     # Get the DataSet part of the file
46     dataset = file.GetDataSet()
47
48     # Ok let's print it !
49     print dataset
50
51     # Use StringFilter to print a particular Tag:
52     sf = gdcm.StringFilter()
53     sf.SetFile(r.GetFile())
54
55     # Check if Attribute exist
56     print dataset.FindDataElement( gdcm.Tag(0x0028,0x0010))
57
58     # Let's print it as string pair:
59     print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))

```

## 12.104 iU22tomultisc.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.

```

=====*/
/*
 * iU22 Raw Data extractor
 */
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
    if( argc < 2 ) return 1;
    // IM_001
    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;
    }

    // * 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 gdcm::File &file = reader.GetFile();
    const gdcm::DataSet &ds = file.GetDataSet();
    const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );
    const gdcm::DataElement &rawdataus = ds.GetDataElement( trawdataus );

    const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
    const gdcm::DataElement &colsrowsframes = 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 &voxelspacing = 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( rawdataus );

    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;
}

```

## 12.105 LargeVRDSExplicit.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 "gdcMAttribute.h"
#include "gdcMFileExplicitFilter.h"
#include "gdcMSequenceOfItems.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 )
            {
                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
        {
            assert( j < npts );
            out.push_back( pts[3*j+0] );
            out.push_back( pts[3*j+1] );
            out.push_back( pts[3*j+2] );
        }
    }
    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];
gdcm::Reader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    return 1;
}

gdcm::File &file = reader.GetFile();
gdcm::DataSet &ds = file.GetDataSet();

gdcm::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/1, 1 ROIContourSequence
gdcm::Tag tag(0x3006,0x0039);

const gdcm::DataElement &roicsq = ds.GetDataElement( tag );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = roicsq.
    GetValueAsSQ();
//sqi->SetNumberOfItems( 1 );
const gdcm::Item &item = sqi->GetItem(1); // Item start at #1
const gdcm::DataSet& nestedds = item.GetNestedDataSet();

gdcm::Tag tcsq(0x3006,0x0040);
if( !nestedds.FindDataElement( tcsq ) )
{
    return 0;
}
const gdcm::DataElement& csq = nestedds.GetDataElement( tcsq );
gdcm::SmartPointer<gdcm::SequenceOfItems> sqi2 = csq.
    GetValueAsSQ();
if( !sqi2 || !sqi2->GetNumberOfItems() )
{
    return 0;
}
//unsigned int nitems = sqi2->GetNumberOfItems();
gdcm::Item &item2 = sqi2->GetItem(1); // Item start at #1

gdcm::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
gdcm::Tag tcontourdata(0x3006,0x0050);
const gdcm::DataElement & contourdata = nestedds2.
    GetDataElement( tcontourdata );
//std::cout << contourdata << std::endl;

//const gdcm::ByteValue *bv = contourdata.GetByteValue();
gdcm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcm::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[0], out.size() / 3, out2);
    //const double *pout = &out[0];
    out = out2;
}

```

```

    out2.clear();
}
assert( out.size() % 3 == 0 );

gdcmm::Attribute<0x3006,0x0050> at_interpolate;
at_interpolate.SetNumberOfValues( (unsigned int)out.size() / 3 );
at_interpolate.SetValues( &out[0], (uint32_t)out.size() );

ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
nesteddds2.Replace( at_interpolate.GetAsDataElement() );
nesteddds2.Replace( ncontourpoints.GetAsDataElement() );

//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;
}

```

## 12.106 MagnifyFile.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 "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"

#include "gdcmmTesting.h"
#include "gdcmmSystem.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 = gdcmm::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;
}

```

## 12.107 MakeTemplate.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 "gdcmFileAnonymizer.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;
}

```

## 12.108 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;
}
}

```

## 12.109 ManipulateFile.py

```

1
14
15 """
16 Usage:
17
18 python ManipulateFile.py input.dcm output.dcm
19
20 Footnote:
21 GDCM 1.2.x would create incorrect Multiframe MR Image Storage file. Try to recover from
22 the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
23 e.g:
24
25 python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
26 """
27
28 import sys
29 import gdcm
30
31 if __name__ == "__main__":
32
33     file1 = sys.argv[1]
34     file2 = sys.argv[2]
35
36     r = gdcm.Reader()
37     r.SetFileName( file1 )
38     if not r.Read():
39         sys.exit(1)
40
41     ano = gdcm.Anonymizer()
42     ano.SetFile( r.GetFile() )
43     ano.RemovePrivateTags()
44     ano.Remove( gdcm.Tag(0x0032,0x1030) )
45     ano.Remove( gdcm.Tag(0x008,0x14) )
46     ano.Remove( gdcm.Tag(0x008,0x1111) )
47     ano.Remove( gdcm.Tag(0x008,0x1120) )
48     ano.Remove( gdcm.Tag(0x008,0x1140) )
49     ano.Remove( gdcm.Tag(0x10,0x21b0) )
50     ano.Empty( gdcm.Tag(0x10,0x10) )
51     ano.Empty( gdcm.Tag(0x10,0x20) )

```

```

52 ano.Empty( gdcm.Tag(0x10,0x30) )
53 ano.Empty( gdcm.Tag(0x20,0x10) )
54 ano.Empty( gdcm.Tag(0x32,0x1032) )
55 ano.Empty( gdcm.Tag(0x32,0x1033) )
56 ano.Empty( gdcm.Tag(0x40,0x241) )
57 ano.Empty( gdcm.Tag(0x40,0x254) )
58 ano.Empty( gdcm.Tag(0x40,0x253) )
59 ano.Empty( gdcm.Tag(0x40,0x1001) )
60 ano.Empty( gdcm.Tag(0x8,0x80) )
61 ano.Empty( gdcm.Tag(0x8,0x50) )
62 ano.Empty( gdcm.Tag(0x8,0x1030) )
63 ano.Empty( gdcm.Tag(0x8,0x103e) )
64 ano.Empty( gdcm.Tag(0x18,0x1030) )
65 ano.Empty( gdcm.Tag(0x38,0x300) )
66 g = gdcm.UIDGenerator()
67 ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
68 ano.Replace( gdcm.Tag(0x0020,0x000d), g.Generate() )
69 ano.Replace( gdcm.Tag(0x0020,0x000e), g.Generate() )
70 ano.Replace( gdcm.Tag(0x0020,0x0052), g.Generate() )
71 #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
72 """
73 ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
74 ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
75 ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
76 ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
77 ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
78 ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
79 ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
80 ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
81 ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
82 ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
83
84 ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
85 ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
86 ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
87 ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
88
89 ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
90
91 ano.Empty( gdcm.Tag(0x0020,0x0020) )
92
93 ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
94
95 #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
96
97 #ano.Empty( gdcm.Tag(0x0028,0x1052) ) #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
Intercept"/>
98 #ano.Empty( gdcm.Tag(0x0028,0x1053) ) #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
Slope"/>
99 #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" ) #<entry group="0028" element="1054" vr="LO" vm="1" name="
Rescale Type"/>
100
101 ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
102 """
103
104 w = gdcm.Writer()
105 w.SetFile( ano.GetFile() )
106 w.SetFileName( file2 )
107 if not w.Write():
108     sys.exit(1)

```

## 12.110 ManipulateSequence.py

```

1
14
15 """
16 Usage:
17
18 python ManipulateSequence.py input.dcm output.dcm
19
20 This was tested using:
21
22 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
23
24 This is a dummy example on how to modify a value set in a nested-nested dataset

```



```

25
26 WARNING:
27 Do not use as-is in production, this is just an example
28 This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
29 """
30
31 import sys
32 import gdcm
33
34 if __name__ == "__main__":
35
36     file1 = sys.argv[1]
37     file2 = sys.argv[2]
38
39     r = gdcm.Reader()
40     r.SetFileName( file1 )
41     if not r.Read():
42         sys.exit(1)
43
44     f = r.GetFile()
45     ds = f.GetDataSet()
46     tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
47     if ds.FindDataElement( tsis ):
48         sis = ds.GetDataElement( tsis )
49         #sqsis = sis.GetSequenceOfItems()
50         # GetValueAsSQ handle more cases
51         sqsis = sis.GetValueAsSQ()
52         if sqsis.GetNumberOfItems():
53             item1 = sqsis.GetItem(1)
54             nestedds = item1.GetNestedDataSet()
55             tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
56             if nestedds.FindDataElement( tprcs ):
57                 prcs = nestedds.GetDataElement( tprcs )
58                 sqprcs = prcs.GetSequenceOfItems()
59                 if sqprcs.GetNumberOfItems():
60                     item2 = sqprcs.GetItem(1)
61                     nestedds2 = item2.GetNestedDataSet()
62                     # (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
63                     tcm = gdcm.Tag(0x0008,0x0104)
64                     if nestedds2.FindDataElement( tcm ):
65                         cm = nestedds2.GetDataElement( tcm )
66                         mystr = "GDCM was here"
67                         cm.SetByteValue( mystr, gdcm.VL( len(mystr) ) )
68
69     w = gdcm.Writer()
70     w.SetFile( f )
71     w.SetFileName( file2 )
72     if not w.Write():
73         sys.exit(1)

```

## 12.111 MergeFile.py

```

1
14
15 """
16 Usage:
17
18 python MergeFile.py input1.dcm input2.dcm
19
20 It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
21 and copy the Stored Pixel values from input2.dcm
22 This script even works when input2.dcm is a Secondary Capture and does not contains information
23 such as IOP and IPP...
24 """
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30
31     file1 = sys.argv[1]
32     file2 = sys.argv[2]
33
34     r1 = gdcm.ImageReader()
35     r1.SetFileName( file1 )

```

```

36  if not r1.Read():
37      sys.exit(1)
38
39  r2 = gdcm.ImageReader()
40  r2.SetFileName( file2 )
41  if not r2.Read():
42      sys.exit(1)
43
44  # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
45  # Instead always prefer to only copy the Raw Data Element.
46  # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
47  r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
48
49  w = gdcm.ImageWriter()
50  w.SetFile( r1.GetFile() )
51  #w.SetImage( r2.GetImage() ) # See comment above
52  w.SetImage( r1.GetImage() )
53
54  w.SetFileName( "merge.dcm" )
55  if not w.Write():
56      sys.exit(1)
57
58  sys.exit(0)

```

## 12.112 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 and 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;
}

```

## 12.113 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;
}
}

```

## 12.114 MIPViewer.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;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.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:vtkgdcm.jar:gdcm.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("vtkgdcmJava");
    }

    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();
    }
}

```

## 12.115 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 Image 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 gdcms;

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;
}
}

```

## 12.116 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], ipzspacing );

// 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();
}
}

```

## 12.117 MPRViewer2.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=vtkgdc.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:vtkgdc.jar:gdcm.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("vtkgdc.jar");
    }

    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 = oy+(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 );
}

```

## 12.118 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.aflTimeConstant[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.ucTOFIInflow                    = 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 gdcm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
    //std::cout << csael << std::endl;

    const gdcm::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 gdcm::CSAHeaderDict &csadict =
        gdcm::Global::GetInstance().GetDicts().
        GetCSAHeaderDict();
    const gdcm::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."<TlmapFuncutor">
    {
        <Class> "<TlmapFuncutor@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;
}

```

## 12.119 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;
}
}

```

## 12.120 NewSequence.py

```

1
14
15 """
16 Usage:
17
18     python NewSequence.py input.dcm output.dcm
19
20
21 Thanks to Robert Irie for code
22 """
23
24 import sys
25 import gdcM
26
27 if __name__ == "__main__":
28
29     file1 = sys.argv[1]
30     file2 = sys.argv[2]
31
32     r = gdcM.Reader()
33     r.SetFileName( file1 )
34     if not r.Read():
35         sys.exit(1)
36
37     f = r.GetFile()
38     ds = f.GetDataSet()
39     #tsis = gdcM.Tag(0x0008,0x2112) # SourceImageSequence
40
41     # Create a dataelement
42     de = gdcM.DataElement(gdcM.Tag(0x0010, 0x2180))
43     de.SetByteValue("Occupation", gdcM.VL(len("Occupation")))
44     de.SetVR(gdcM.VR(gdcM.VR.SH))
45
46     # Create an item
47     it=gdcM.Item()
48     it.SetVLToUndefined() # Needed to not popup error message
49     #it.InsertDataElement(de)
50     nds=it.GetNestedDataSet()
51     nds.Insert(de)
52
53     # Create a Sequence
54     sq=gdcM.SequenceOfItems().New()
55     sq.SetLengthToUndefined()
56     sq.AddItem(it)
57
58     # Insert sequence into data set
59     des=gdcM.DataElement(gdcM.Tag(0x0400,0x0550))
60     des.SetVR(gdcM.VR(gdcM.VR.SQ))
61     des.SetValue(sq.__ref__())
62     des.SetVLToUndefined()
63
64     ds.Insert(des)
65
66     w = gdcM.Writer()

```

```

67  w.SetFile( f )
68  w.SetFileName( file2 )
69  if not w.Write():
70      sys.exit(1)

```

## 12.121 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"

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();
    #if (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();
    #if (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();
    #if (VTK_MAJOR_VERSION >= 6)
    wr->SetInputConnection( w2if->GetOutputPort() );
    #else
    wr->SetInput( w2if->GetOutput() );
    #endif
    wr->SetFileName ( "offscreenimage.png" );
    wr->Write();

    reader->Delete();
    renWin->Delete();
    renderer->Delete();
    windowlevel->Delete();
    actor->Delete();
    w2if->Delete();
    wr->Delete();

    return 0;
}

```

## 12.122 PatchFile.cxx

This is a C++ example on how to use [gdcm::Attribute](#)

```

/*=====
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 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 "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        return 1;
    }
    const char *f = argv[1];
    const char *out = argv[2];
    gdcm::Reader r;
    r.SetFileName( f );
    if( !r.Read() )
    {
        return 1;
    }

    gdcm::File &file = r.GetFile();
    gdcm::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
//
{
    gdcm::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;
}

```

## 12.123 PhilipsPrivateRescaleInterceptSlope.py

```

1
14
15 """
16 Usage:
17
18 python
19 """
20
21 import gdcmm
22 import sys
23
24 filename = sys.argv[1]
25 tmpfile = "/tmp/philips_rescaled.dcm"
26
27
28 # Need to access some private tags, read the file :
29 reader = gdcmm.Reader()
30 reader.SetFileName( filename )
31 if not reader.Read():
32     sys.exit(1)
33
34 ds = reader.GetFile().GetDataSet()
35
36 #print ds
37 # (2005,1409)      DS      4      0.0
38 # (2005,140a)      DS     16     1.52283272283272
39
40 # (2005,0014)      LO     26     Philips MR Imaging DD 005
41 tag1 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcmm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
43 print tag1
44 print tag2
45
46 # make sure to do a copy, we want the private tag to remain
47 # otherwise gdcmm gives us a reference
48 e11 = gdcmm.DataElement( ds.GetDataElement( tag1 ) )
49 print e11
50 e12 = gdcmm.DataElement( ds.GetDataElement( tag2 ) )
51 print e12
52
53 # (0028,1052) DS [-1000]          # 6, 1 RescaleIntercept
54 # (0028,1053) DS [1]             # 2, 1 RescaleSlope
55
56 e11.SetTag( gdcmm.Tag(0x0028,0x1052) )
57 e12.SetTag( gdcmm.Tag(0x0028,0x1053) )
58
59 ds.Insert( e11 )
60 ds.Insert( e12 )
61
62 w = gdcmm.Writer()
63 w.SetCheckFileMetaInformation( False )
64 w.SetFileName( tmpfile )
65 w.SetFile( reader.GetFile() )
66 if not w.Write():
67     sys.exit(1)
68
69 print "success"

```

## 12.124 PlaySound.py

```

1
14
15 """
16 Usage:
17
18 python PlaySound.py input.dcm
19 """
20
21 import gdcmm
22 import sys
23
24 #filename = "/home/mmalaterre/Creatis/gdcmmDataExtra/gdcmmNonImageData/audio_from_rafael_sanguinetti.dcm"

```

```

25 filename = sys.argv[1]
26 print filename
27
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31     sys.exit(1)
32
33 ds = r.GetFile().GetDataSet()
34
35 waveformtag = gdcm.Tag(0x5400,0x0100)
36 waveformsq = ds.GetDataElement( waveformtag )
37 #print waveformsq
38
39 #print dir(waveformsq)
40
41 items = waveformsq.GetSequenceOfItems()
42
43 if not items.GetNumberOfItems():
44     sys.exit(1)
45
46 item = items.GetItem(1)
47 #print item
48
49 waveformds = item.GetNestedDataSet()
50 #print waveformds
51
52 waveformdatatag = gdcm.Tag(0x5400,0x1010)
53 waveformdata = waveformds.GetDataElement( waveformdatatag )
54
55 #print waveformdata.GetPointer()
56 bv = waveformdata.GetByteValue()
57 print dir(bv)
58
59 #print bv.GetPointer()
60 print bv.GetLength()
61 l = 116838
62
63 file='test.wav'
64 myfile = open(file, "wb")
65 s = bv.GetPointer()
66 for i in range(0, l):
67     myfile.write(s[i])
68 myfile.close()
69
70 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
71 if sys.platform.startswith('win'):
72     from winsound import PlaySound, SND_FILENAME, SND_ASYNC
73     PlaySound(file, SND_FILENAME|SND_ASYNC)
74 elif sys.platform.find('linux')>-1:
75     from wave import open as waveOpen
76     from ossaudiodev import open as ossOpen
77     s = waveOpen(file,'rb')
78     (nc,sw,fr,nf,comptype, compname) = s.getparams( )
79     dsp = ossOpen('/dev/dsp','w')
80     try:
81         from ossaudiodev import AFMT_S16_NE
82     except ImportError:
83         if byteorder == "little":
84             AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
85         else:
86             AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
87     dsp.setparameters(AFMT_S16_NE, nc, fr)
88     data = s.readframes(nf)
89     s.close()
90     dsp.write(data)
91     dsp.close()

```

## 12.125 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 (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:
 * Jean-Pierre Roux for providing the sample datasets
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.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 );

    assert( data_size != outputlen );
    if( data_size == outputlen )
    {
        return;
    }
    typedef unsigned char byte;
    enum {
        COLORMODE = 0x81,
        ESCMODE = 0x82,
        REPEATMODE = 0x83
    };

    byte* src = (byte*)data_in;
    byte* dest = (byte*)&new_stream[0];
    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 ther other (RGB <-> GRAY)
    while (ps)
    {
        // next byte:
        byte b = *src++;
        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 swithing 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( strcmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
        isrle = true;
        return 1;
    }
    if( strcmp( 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[0], (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;
}

```

## 12.126 PrivateDict.py

```

1
14
15 """
16 """
17
18 import gdcmm
19 import sys,os
20
21 if __name__ == "__main__":
22     #gdcmm.Trace.DebugOn()
23     globInst = gdcmm.Global.GetInstance()
24     # Try to load Part3.xml file
25     # This file is too big for being accessible directly at runtime.
26     globInst.LoadResourcesFiles()
27
28
29     # Get a private tag from the runtime dicts. LoadResourcesFiles could
30     # have failed but this has no impact on the private dict
31
32     d = globInst.GetDicts()
33     print d.GetDictEntry( gdcmm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
34     pd = d.GetPrivateDict()
35     print pd.GetDictEntry( gdcmm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )

```

## 12.127 PublicDict.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 example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcmmGlobal.h"
#include "gdcmmDicts.h"
#include "gdcmmDict.h"
#include "gdcmmCSAHeader.h"
#include "gdcmmPrivateTag.h"

int main(int , char *[])
{
    const gdcmm::Global& g = gdcmm::Global::GetInstance(); // sum of all
        knowledge !
    const gdcmm::Dicts &dicts = g.GetDicts();
    const gdcmm::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:
gdcmm::Tag patient_name(0x10,0x10);
const gdcmm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
std::cout << entry1 << std::endl;

// 2. From all dicts:
const gdcmm::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 = 0;
const gdcmm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
std::cout << entry3 << std::endl;

// Private attributes:

// try with a private tag now:
const gdcmm::PrivateTag &private_tag =
    gdcmm::CSAHeader::GetCSAImageHeaderInfoTag();
//std::cout << private_tag << std::endl;
const gdcmm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.
    GetOwner());
std::cout << entry4 << std::endl;

// Let's pretend that private lookup is on 0x10xx elements:
gdcmm::PrivateTag dummy = private_tag;
dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
const gdcmm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.
    GetOwner());
std::cout << entry5 << std::endl;

return 0;
}

```

## 12.128 QIDO-RS.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 "gdcmmJSON.h"

/*
 * Simple QIDO-RS round-trip to test implementation of gdcmm::JSON
 * See Supl66 for details
 */
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::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;
}

```

## 12.129 ReadAndDumpDICOMDIR.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 read and dump a DICOMDIR File
 *
 * Thanks:
 * Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmReader.h"
#include "gdcmMediaStorage.h"

typedef std::set<gdcm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

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 << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();
    gdcm::FileMetaInformation &fmi = file.GetHeader();

    gdcm::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcm::MediaStorage::MediaStorageDirectoryStorage
        )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }
}

```

```

if (fmi.FindDataElement( gdcm::Tag (0x0002, 0x0002)))
{
    strm.str("");
    fmi.GetDataElement( gdcm::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()==gdcm::Tag (0x0004, 0x1220))
    {
        const gdcm::DataElement &de = (*it);
        // ne pas utiliser GetSequenceOfItems pour extraire les items
        gdcm::SmartPointer<gdcm::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 << " " << strm.str() << std::endl;
                }
            }
        }
    }
}

```

```

        //STUDY DATE
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
        std::cout << "          STUDY DATE : " << strm.str() << std::endl;

        //STUDY DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x1030)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0020, 0x000e)))
                sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
            std::cout << "          SERIE UID" << strm.str() << std::endl;

            //SERIE MODALITY
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x0060)))
                sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
            std::cout << "          SERIE MODALITY" << strm.str() << std::endl;

            //SERIE DESCRIPTION
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x103e)))
                sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
                sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1511)))
                    sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1500)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

    }
}
}
}
itemused++;
}
}
}
return 0;
}

```

## 12.130 ReadAndDumpDICOMDIR.py

```

1
23
24
25
26 import sys
27 import gdcmm
28
29 if __name__ == "__main__":
30     # Check arguments
31     if (len(sys.argv) < 2):
32         # No filename passed
33         print "No input filename found"
34         quit()
35
36     filename = sys.argv[1]
37
38
39     # Read file
40     reader = gdcmm.Reader()
41     reader.SetFileName(filename)
42     if (not reader.Read()):
43         print "Unable to read %s" % (filename)
44         quit()
45
46     file = reader.GetFile()
47
48     # Retrieve header information
49     fileMetaInformation = file.GetHeader()
50     print fileMetaInformation
51
52     # Retrieve data set
53     dataSet = file.GetDataSet()
54     #print dataSet
55
56     # Check media storage
57     mediaStorage = gdcmm.MediaStorage()
58     mediaStorage.SetFromFile(file)
59     if (gdcmm.MediaStorage.GetMSType(str(mediaStorage)) !=
gdcmm.MediaStorage.MediaStorageDirectoryStorage):
60         # File is not a DICOMDIR
61         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))

```



```

62         quit()
63
64     # Check Media Storage SOP Class
65     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
66         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
67         # Check SOP UID
68         if (sopClassUid != "1.2.840.10008.1.3.10"):
69             # File is not a DICOMDIR
70             print "This file is not a DICOMDIR"
71     else:
72         # Not present
73         print "Media Storage SOP Class not present"
74         quit()
75
76     # Iterate through the DICOMDIR data set
77     iterator = dataSet.GetDES().begin()
78     while (not iterator.equal(dataSet.GetDES().end())):
79         dataElement = iterator.next()
80
81         # Check the element tag
82         if (dataElement.GetTag() == gdcm.Tag(0x0004, 0x1220)):
83             # The 'Directory Record Sequence' element
84             sequence = dataElement.GetValueAsSQ()
85
86             # Loop through the sequence items
87             itemNr = 1
88             while (itemNr < sequence.GetNumberOfItems()):
89                 item = sequence.GetItem(itemNr)
90
91                 # Check the element tag
92                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
93                     # The 'Directory Record Type' element
94                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
95
96                     # PATIENT
97                     while (value.strip() == "PATIENT"):
98                         print value.strip()
99                         # Print patient name
100                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
101                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
102                             print value
103
104                         # Print patient ID
105                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
106                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
107                             print value
108
109                         # Next
110                         itemNr = itemNr + 1
111                         item = sequence.GetItem(itemNr)
112                         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
113                             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
114
115                         # STUDY
116                         while (value.strip() == "STUDY"):
117                             print value.strip()
118
119                             # Print study UID
120                             if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
121                                 value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
122                                 print value
123
124                             # Print study date
125                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
126                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
127                                 print value
128
129                             # Print study description
130                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
131                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
132                                 print value
133
134                             # Next
135                             itemNr = itemNr + 1
136                             item = sequence.GetItem(itemNr)
137                             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
138                                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).

```

```

139
140         # SERIES
141         while (value.strip() == "SERIES"):
142             print value.strip()
143
144         # Print series UID
145         if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
146             value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e))).
147         GetValue()
148             print value
149
150         # Print series modality
151         if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
152             value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060))).
153         GetValue()
154             print "Modality"
155             print value
156
157         # Print series description
158         if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
159             value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e))).
160         GetValue()
161             print "Description"
162             print value
163
164         # Next
165         itemNr = itemNr + 1
166         item = sequence.GetItem(itemNr)
167         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
168             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430))).
169         GetValue()
170
171         # IMAGE
172         while (value.strip() == "IMAGE"):
173             print value.strip()
174
175         # Print image UID
176         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
177             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511))).
178         GetValue()
179             print value
180
181         # Next
182         if (itemNr < sequence.GetNumberOfItems()):
183             itemNr = itemNr + 1
184         else:
185             break
186
187         item = sequence.GetItem(itemNr)
188         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
189             value = str(item.GetDataElement(
190                 gdcm.Tag(0x0004, 0x1430)).GetValue())
191
192         # Next
193         itemNr = itemNr + 1

```

## 12.131 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 "gdcmReader.h"
#include "gdcmAttribute.h"
#include "gdcmDirectory.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 &operator = (const 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
    //

    gdcm::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
    gdcm::MediaStorage media_storage;
    media_storage.SetFromFile(file);
    if(media_storage != gdcm::MediaStorage::MediaStorageDirectoryStorage
    ) {
        throw std::runtime_error("This file is not a DICOMDIR");
    }

    auto media_storage_sop_class_uid = get_string(file_meta_information,
        gdcm::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
    //

    gdcm::VL first_item_offset = 0;
    auto it = data_set.Begin();
    for(; it != data_set.End() && it->GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE); ++it)
    {
        first_item_offset += it->GetLength<gdcm::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 lenght = VR field lenght,
    // 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() != gdcm::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<
gdcmm::ExplicitDataElement>();

    // Child offset
    gdcmm::Attribute<TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET>
child_offset;
    child_offset.SetFromDataElement(item.GetDataElement(
gdcmm::Tag (TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET)));

    // Sibling offset
    gdcmm::Attribute<TAG_NEXT_DIRECTORY_RECORD_OFFSET>
sibling_offset;
    sibling_offset.SetFromDataElement(item.GetDataElement(
gdcmm::Tag (TAG_NEXT_DIRECTORY_RECORD_OFFSET)));

    // Record Type
    auto record_type = trim(get_string(item, gdcmm::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, gdcmm::Tag (TAG_PATIENTS_NAME));
        patient->id = get_string(item, gdcmm::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, gdcmm::Tag (TAG_STUDY_DATE));
        study->description = get_string(item, gdcmm::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, gdcmm::Tag (TAG_MODALITY));
        series->description = get_string(item, gdcmm::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, gdcmm::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;
}

```

## 12.132 ReadAndPrintAttributes.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 small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.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];

    // Instanciate 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 &dicts = g.GetDicts();
    const gdcm::Dict &pubdict = dicts.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 &del =
    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.FindDataElement( 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;
    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 iieee double)
    Attribute<0x3004,0x000e>::ArrayType v = at.
        GetValue();
    std::cout << "DoseGridScaling=" << v << std::endl;
}

return 0;
}

```

## 12.133 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 "gdcmReader.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.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(0x01,0x42,"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;
}

```

## 12.134 ReadFiles.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 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);
}
}
}

```

## 12.135 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 "gdcmReader.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"

#include <iostream>
#include <string>

using namespace gdcm;

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 InternalSDODataset.size();
    }
    void AddSDOElement(SDOElement const &sdoelement) {
        InternalSDODataset.push_back( sdoelement );
    }
    const SDOElement &GetSDOElement(SizeType index) const {
        return InternalSDODataset[index];
    }
    const SDOElement &GetSDOElementByName(const char *) const {
        return InternalSDODataset[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;
}

```

## 12.136 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)
{
    std::cerr << "BAD ALLOC Exception caught!" << std::endl;
}
catch (...)
{
    std::cerr << "Exception caught!" << std::endl;
}
}

return 0;
}

```

## 12.137 ReadSeriesIntoVTK.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.

=====*/
// We are required to call the package 'vtk' eventhough I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcM.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdcM.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("vtkgdcMJava");
        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() );
}

```

## 12.138 ReadUTF8QtDir.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.
=====*/

/*
 * 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 "gdcmReader.h"
#include "gdcmDirectory.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;
    }
    gdcm::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;
    gdcm::Directory dir;
    unsigned int nfiles = dir.Load( dirname, true );
    const gdcm::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;
}

```

## 12.139 RefCounting.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;

/*
 * 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;
    }
}

```

## 12.140 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://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:
 * $ 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 pupose 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;
}
}

```

## 12.141 RemovePrivateTags.py

```

1
14
15 """
16 Usage:
17
18 python RemovePrivateTags.py input.dcm output.dcm
19 """
20
21 import sys
22 import gdcm
23
24
25 if __name__ == "__main__":
26
27     file1 = sys.argv[1]
28     file2 = sys.argv[2]
29
30     # Instantiate the reader.
31     r = gdcm.Reader()
32     r.SetFileName( file1 )
33     if not r.Read():
34         sys.exit(1)
35
36     # Remove private tags
37     ano = gdcm.Anonymizer()
38     ano.SetFile( r.GetFile() )
39     if not ano.RemovePrivateTags():
40         sys.exit(1)
41
42     # Write DICOM file
43     w = gdcm.Writer()
44     w.SetFile( ano.GetFile() )
45     #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
46     w.SetFileName( file2 )
47     if not w.Write():
48         sys.exit(1)
49
50     # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect)
    DICOM file
51     # (application level)

```

## 12.142 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.ComputeInterceptSlopePixelType() );

            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;
        }
    }
}

```

## 12.143 reslicesphere.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.

=====*/

```

```

//
// 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 <sstream>
#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 "gdcmDirectory.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.GetFileNames();
    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();

    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());
    #if (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());
}

```

```

        _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;
}

```

## 12.144 ReWriteSCAsMR.py

```

1
14
15 """
16 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
17 Slope/Intercept
18 and saving the Pixel Spacing in (0028,0030)
19 """
20 import gdcm
21 import sys,os
22
23 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
24     ds = r.GetFile().GetDataSet()
25     # Check Source Image Sequence
26     if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
27         sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
28         sqsis = sis.GetSequenceOfItems()
29         if sqsis.GetNumberOfItems():
30             item1 = sqsis.GetItem(1)
31             nestedds = item1.GetNestedDataSet()

```

```

32     if nestedds.FindDataElement( gdcmm.Tag(0x0008,0x1150) ):
33         ReferencedSOPClassUID = nestedds.GetDataElement( gdcmm.Tag(0x0008,0x1150) )
34         raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
35         uids = gdcmm.UIDs()
36         # what is the actual object we are looking at ?
37         ms = gdcmm.MediaStorage()
38         ms.SetFromDataSet(ds)
39         msuid = ms.GetString()
40         uids.SetFromUID( msuid )
41         msuidname = uids.GetName() # real Media Storage Name
42         uids.SetFromUID( raw )
43         sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
44         # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is
         correct
45         if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
46             return True
47         # in all other case simply return the currentspacing:
48         return False
49
50 if __name__ == "__main__":
51     r = gdcmm.ImageReader()
52     filename = sys.argv[1]
53     r.SetFileName( filename )
54     if not r.Read():
55         sys.exit(1)
56     f = r.GetFile()
57
58     if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
59         # Special handling of the spacing:
60         # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture
        Image Storage'
61         # while we would rather have 'MR Image Storage'
62         gdcmm.ImageHelper.SetForcePixelSpacing( True )
63         mrspacing = gdcmm.ImageHelper.GetSpacingValue( r.GetFile() )
64         # TODO: I cannot do simply the following:
65         #image.SetSpacing( mrspacing )
66         image.SetSpacing(0, mrspacing[0] )
67         image.SetSpacing(1, mrspacing[1] )
68         image.SetSpacing(2, mrspacing[2] )
69         gdcmm.ImageHelper.SetForceRescaleInterceptSlope( True )
70         ris = gdcmm.ImageHelper.GetRescaleInterceptSlopeValue(
            r.GetFile() )
71         image.SetIntercept( ris[0] )
72         image.SetSlope( ris[1] )
73
74     outfilename = sys.argv[2]
75     w = gdcmm.ImageWriter()
76     w.SetFileName( outfilename )
77     w.SetFile( r.GetFile() )
78     w.SetImage( image )
79     if not w.Write():
80         sys.exit(1)
81
82     sys.exit(0)

```

## 12.145 rle2img.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_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 (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:
* Mauro Maiorca for bringing to our attention on this new ELSCINT1
* compression algorithm : PMSCT_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 "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.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;
            //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)(temp[i] + delta);
            output.push_back( value );
            delta = value;
        }
        //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 << argv[0] << "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;

    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<0x0028,0x0010> at1;
    at1.SetFromDataSet( ds );
    gdcm::Attribute<0x0028,0x0011> at2;
    at2.SetFromDataSet( ds );

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
    pixeldata.SetVR( gdcm::VR::OW );
    gdcm::VL bv2l = bv2->GetLength();
    gdcm::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[0], (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
    reader.GetFile().GetDataSet().Replace( pixeldata );

    reader.GetFile().GetHeader().SetDataSetTransferSyntax(

```

```

    gdcmm::TransferSyntax::ExplicitVRLittleEndian);
gdcmm::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 gdcmm::Anonymizer::RemovePrivateTags if needed...
writer.GetFile().GetDataSet().Remove( compressionpixeldata.
    GetTag() );
std::string outfilename;
if (argc > 2)
    outfilename = argv[2];
else
    outfilename = "out.rle.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;
}

```

## 12.146 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"

/*
 * 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;
}

```

## 12.147 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;
    }
}

```

## 12.148 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 r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
            byte[] rbuf = new byte[ (int)r1 ];
            long r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
            assert r1 == r12;
            long g1 = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
            byte[] gbuf = new byte[ (int)g1 ];
            long g12 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
            assert g1 == g12;
            long b1 = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
            byte[] bbuf = new byte[ (int)b1 ];
            long b12 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
            assert b1 == b12;
            colorModel = new IndexColorModel(8, (int)r1, 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" );
}
}

```

## 12.149 ScanDirectory.py

```

1
14
15 import gdcmm
16 import sys,os
17
18 class ProgressWatcher(gdcm.SimpleSubjectWatcher):
19     def ShowProgress(self, sender, event):
20         pe = gdcm.ProgressEvent.Cast(event)
21         print pe.GetProgress()
22     def EndFilter(self):
23         print "Yay ! I am done"
24
25 if __name__ == "__main__":
26     directory = sys.argv[1]

```

```

27
28 # Define the set of tags we are interested in
29 t1 = gdcm.Tag(0x8,0x8);
30 t2 = gdcm.Tag(0x10,0x10);
31
32 # Iterate over directory
33 d = gdcm.Directory();
34 nfiles = d.Load( directory );
35 if(nfiles == 0): sys.exit(1);
36 # System.Console.WriteLine( "Files:\n" + d.toString() );
37
38 filenames = d.GetFilenames()
39
40 # Get rid of any Warning while parsing the DICOM files
41 gdcm.Trace.WarningOff()
42
43 # instanciate Scanner:
44 sp = gdcm.Scanner.New();
45 s = sp.__ref__()
46 w = ProgressWatcher(s, 'Watcher')
47
48 s.AddTag( t1 );
49 s.AddTag( t2 );
50 b = s.Scan( filenames );
51 if(not b): sys.exit(1);
52
53 print "success" ;
54 #print s
55
56 pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
57 pttv.Start()
58 # iterate until the end:
59 while( not pttv.IsAtEnd() ):
60     # get current value for tag and associated value:
61     # if tag was not found, then it was simply not added to the internal std::map
62     # Warning value can be None
63     tag = pttv.GetCurrentTag()
64     value = pttv.GetCurrentValue()
65     print tag,"->",value
66     # increment iterator
67     pttv.Next()
68
69 sys.exit(0)

```

## 12.150 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;
}
}

```

## 12.151 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( 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;
}
}

```

## 12.152 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;
            }
        }
    }
}

```

```

    gdcm.File file = reader.GetFile();

    gdcm.StringFilter filter = new gdcm.StringFilter();
    filter.SetFile(file);
    string value = filter.ToString(new gdcm.Tag(0x0010, 0x0010));

    Console.WriteLine("Patient Name: " + value);
    return 0;
}
}
}

```

## 12.153 SimpleScanner.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.
=====*/

/*
 * 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 gdcm::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
 * gdcm::System::FileExists to decide whether or not the file actually exist on the disk.
 *
 * It was tested on this particular image:
 * ./SimpleScanner gdcmData/012345.002.050.dcm
 */

#include "gdcmStrictScanner.h"
#include "gdcmSimpleSubjectWatcher.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)
    {
        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;
}

```

## 12.154 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 );

gdcmm::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 );

gdcmm::Scanner s;
s.AddTag( gdcmm::Tag(0x20,0x32) ); // Image Position (Patient)
//s.AddTag( gdcmm::Tag(0x20,0x37) ); // Image Orientation (Patient)
s.Scan( dir.GetFileNames() );

//s.Print( std::cout );

// Count how many different IPP there are:
const gdcmm::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;
}

```

## 12.155 SortImage.py

```

1
14
15 """
16 Usage:
17
18 python SortImage.py dirname
19 """
20
21 import gdcmm
22 import sys
23
24 def PrintProgress(object, event):
25     assert event == "ProgressEvent"
26     print "Progress:", object.GetProgress()
27
28 def MySort(ds1, ds2):
29     # compare ds1
30     return False
31
32 if __name__ == "__main__":
33
34     dirname = sys.argv[1]
35     d = gdcmm.Directory()
36     d.Load( dirname )
37
38     print d
39
40     sorter = gdcmm.Sorter()
41     sorter.SetSortFunction( MySort )
42     #sorter.AddObserver( "ProgressEvent", PrintProgress )
43     sorter.Sort( d.GetFileNames() )
44
45     print "Sorter:"
46     print sorter

```

## 12.156 SortImage2.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/SortImage.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

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;
    }
}

```

## 12.157 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;
}

```

## 12.158 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 "gdcmJPEGLSCodec.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) );

```

```

del.SetVR( gdc::VR::UI );
gdc::MediaStorage ms(
    gdc::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()));
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdc::DataElement de2( gdc::Tag(0x28,0x04) );
//de.SetTag(gdc::Tag(0x28,0x04));
de2.SetVR( gdc::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdc::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdc::Attribute<0x0028,0x0010> row = {extent[0]/a};
ds.Insert( row.GetAsDataElement() );

gdc::Attribute<0x0028,0x0011> col = {extent[1]/a};
ds.Insert( col.GetAsDataElement() );

gdc::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdc::Attribute<0x0028,0x0002> at1 = {1};
ds.Insert( at1.GetAsDataElement() );

gdc::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdc::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*
ds1.Remove( gdc::Tag(0x0028,0x0008) );

gdc::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 = gdc::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);

    gdcm::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, outfile, resolution))
        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));
    assert( of && !of.eof() && of.good() );
    of.write(tmpBuffer2, theBufferSize1);
    of.flush();
    assert( of );

    return 0;
}

```

## 12.159 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( gdcm::DASComp( 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( gdcm::TMComp( date+datelen, timelen ) );
    ds.Insert( contenttime.GetAsDataElement() );

    gdcm::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
    // gdcm will pick the Word in case Byte class is not compatible:
    gdcm::MediaStorage ms =
        gdcm::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() );
//      gdcm::MediaStorage msl = w.ComputeTargetMediaStorage();
//      std::cout << msl << std::endl;
std::cout << ds << std::endl;
gdcm::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;
}

```

## 12.160 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 <string.h> // memcpy

int myfunc()
{
    char vl_str[4];
    const char raw[] = "\000\000\000\004";
    memcpy(vl_str, raw, 4);
    uint32_t vl;
    gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem(
        ((uint32_t*)(&vl_str)), gdcm::SwapCode::BigEndian, 1);
    memcpy(&vl, vl_str, 4);
    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 };
    gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
        (array,
         gdcm::SwapCode::BigEndian,1);
    if ( array[0] != 0x3412 )
    {
        std::cerr << std::hex << "array: " << array[0] << std::endl;
        return 1;
    }

    return 0;
}

```

## 12.161 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;
}

```

## 12.162 TestReader.py

This is a C++ example on how to use `gdcm::Reader`

```

1
14
15 import os,sys
16 import gdcm
17
18 def TestRead(filename, verbose = False):
19     r = gdcm.Reader()
20     r.SetFileName( filename )
21     success = r.Read()
22     #if verbose: print r.GetFile()
23     if verbose: print (r.GetFile().GetDataSet())
24     return success
25
26 if __name__ == "__main__":
27     success = 0
28     try:
29         filename = os.sys.argv[1]
30         success += TestRead( filename, True )
31     except:
32         # loop over all files:
33         gdcm.Trace.DebugOff()
34         gdcm.Trace.WarningOff()
35         t = gdcm.Testing()
36         nfiles = t.GetNumberOfFileNames()
37         for i in range(0,nfiles):

```



```

38         filename = t.GetFileName(i)
39         success += TestRead( filename )
40
41
42     # Test succeed ?
43     sys.exit(success == 0)

```

## 12.163 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 <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->scalarpainter;
    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

    gdcm::ImageReader reader;
    reader.SetFileName( reference );
    if( !reader.Read() )
    {
        // That would be very bad...
        assert(0);
    }

    const gdcm::Image &image = reader.GetImage();
    gdcm::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/threadgdcm.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 && gdcm::System::FileIsDirectory( argv[1] ) )
    {
        gdcm::Directory d;
        d.Load( argv[1] );
        gdcm::Directory::FileNamesType l = d.
            GetFileNames();
        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;
}

```

## 12.164 TraverseModules.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 "gdcmGlobal.h"
#include "gdcmIODs.h"
#include "gdcmIOD.h"
#include "gdcmMacros.h"
#include "gdcmIODEntry.h"
#include "gdcmModules.h"
#include "gdcmModule.h"

```

```

#include "gdcmanonymizer.h"
#include "gdcmdicts.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;
}

```

## 12.165 uid\_unique.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 "gdcmuuidgenerator.h"

```

```

#include <iostream>
#include <string>
#include <set>

int main()
{
    gdcm::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(1)
    {
        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 );
    }
}

```

## 12.166 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 );

```

```

    gdcmm::Attribute<0x0020,0x000e> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort3(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    // This is a floating point number is the comparison ok ?
    gdcmm::Attribute<0x0020,0x0037> at1;
    at1.Set( ds1 );
    gdcmm::Attribute<0x0020,0x0037> at2;
    at2.Set( ds2 );
    return at1 < at2;
}

bool mysort4(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    // Do the IPP sorting here
    gdcmm::Attribute<0x0020,0x0032> ipp1;
    gdcmm::Attribute<0x0020,0x0037> iop1;
    ipp1.Set( ds1 );
    iop1.Set( ds1 );
    gdcmm::Attribute<0x0020,0x0032> ipp2;
    gdcmm::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 = gdcmm::Testing::GetDataExtraRoot();
    std::string dirl;
    if( argc < 2 )
    {
        if( !extradataroot )
        {
            return 1;
        }
        dirl = extradataroot;
        dirl += "/gdcmmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dirl = argv[1];
    }

    gdcmm::Directory d;
    d.Load( dirl.c_str(), true ); // recursive !
    const gdcmm::Directory::FileNamesType &ll = d.
        GetFileNames();
    const size_t nfiles = ll.size();
    std::cout << nfiles << std::endl;

    //if( nfiles != 280 )
    // {
    //     return 1;
    // }

    //d.Print( std::cout );

    gdcmm::Scanner s0;
    const gdcmm::Tag t1(0x0020,0x000d); // Study Instance UID

```

```

const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
//const gdcm::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.GetFileNames() );
if( !b )
{
    std::cerr << "Scanner failed" << std::endl;
    return 1;
}

//s0.Print( std::cout );

// Only get the DICOM files:
gdcm::Directory::FileNamesType l2 = s0.GetKeys();
const size_t nfiles2 = l2.size();
std::cout << nfiles2 << std::endl;

if ( nfiles2 > nfiles )
{
    return 1;
}

gdcm::Sorter sorter;
sorter.SetSortFunction( mysort1 );
sorter.StableSort( l2 );

sorter.SetSortFunction( mysort2 );
sorter.StableSort( sorter.GetFileNames() );

sorter.SetSortFunction( mysort3 );
sorter.StableSort( sorter.GetFileNames() );

sorter.SetSortFunction( mysort4 );
sorter.StableSort( sorter.GetFileNames() );

//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;
{
    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFileNames() );

    //s.Print( std::cout );

    const gdcm::Scanner::ValuesType &values = s.GetValues();
    nvalues = values.size();
    std::cout << "There are " << nvalues << " different type of values" << std::endl;
    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;
}

```

## 12.167 WriteBuffer.py

```

1
14
15 """
16 Usage:
17
18 http://chuckhahm.com/Ischem/Zurich/XX_0134
19
20 (2005,1132) SQ (Sequence with undefined length #=8)      # u/1, 1 Unknown Tag & Data
21 (ffff,e000) na (Item with undefined length #=9)          # u/1, 1 Item
22 (2005,0011) LO [Philips MR Imaging DD 002]              # 26, 1 PrivateCreator
23 (2005,1137) PN [PDF_CONTROL_GEN_PARS]                   # 20, 1 Unknown Tag & Data
24 (2005,1138) PN (no value available)                     # 0, 0 Unknown Tag & Data
25 (2005,1139) PN [IEEE_PDF]                              # 8, 1 Unknown Tag & Data
26 (2005,1140) PN (no value available)                     # 0, 0 Unknown Tag & Data
27 (2005,1141) PN (no value available)                     # 0, 0 Unknown Tag & Data
28 (2005,1143) SL 3103                                     # 4, 1 Unknown Tag & Data
29 (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown
    Tag & Data
30 (2005,1147) CS [Y]                                      # 2, 1 Unknown Tag & Data
31 (ffff,e00d) na (ItemDelimitationItem)                   # 0, 0 ItemDelimitationItem
32 (ffff,e000) na (Item with undefined length #=9)          # u/1, 1 Item
33 (2005,0011) LO [Philips MR Imaging DD 002]              # 26, 1 PrivateCreator
34 (2005,1137) PN [PDF_CONTROL_PREP_PARS]                   # 22, 1 Unknown Tag & Data
35 (2005,1138) PN (no value available)                     # 0, 0 Unknown Tag & Data
36 (2005,1139) PN [IEEE_PDF]                              # 8, 1 Unknown Tag & Data
37 (2005,1140) PN (no value available)                     # 0, 0 Unknown Tag & Data
38 (2005,1141) PN (no value available)                     # 0, 0 Unknown Tag & Data
39 (2005,1143) SL 7934                                     # 4, 1 Unknown Tag & Data
40 (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown
    Tag & Data
41 (2005,1147) CS [Y]                                      # 2, 1 Unknown Tag & Data
42 (ffff,e00d) na (ItemDelimitationItem)                   # 0, 0 ItemDelimitationItem
43 ...
44 """
45
46 import sys
47 import gdcm
48
49 if __name__ == "__main__":
50
51     file1 = sys.argv[1]
52     file2 = sys.argv[2]
53
54     r = gdcm.Reader()
55     r.SetFileName( file1 )
56     if not r.Read():
57         sys.exit(1)
58
59     fg = gdcm.FileNameGenerator()
60     f = r.GetFile()
61     ds = f.GetDataSet()
62     tsis = gdcm.Tag(0x2005,0x1132) #
63     if ds.FindDataElement( tsis ):
64         sis = ds.GetDataElement( tsis )
65         #sqsis = sis.GetSequenceOfItems()
66         # GetValueAsSQ handle more cases
67         sqsis = sis.GetValueAsSQ()
68         if sqsis.GetNumberOfItems():
69             nitems = sqsis.GetNumberOfItems();
70             fg.SetNumberOfFileNames( nitems )
71             fg.SetPrefix( file2 )
72             if not fg.Generate():
73                 print "problem"
74                 sys.exit(1)
75             for i in range(0,nitems):
76                 item1 = sqsis.GetItem(i+1) # Item start at 1
77                 nestedds = item1.GetNestedDataSet()
78                 tprcs = gdcm.Tag(0x2005,0x1144) #
79                 if nestedds.FindDataElement( tprcs ):

```

```
80         prcs = nestedds.GetDataElement( tprcs )
81         bv = prcs.GetByteValue()
82         print bv
83         f = open( fg.GetFilename(i) , "w" )
84         f.write( bv.WriteBuffer() )
```

# Index

- ~ASN1
  - gdcmm::ASN1, [131](#)
- ~AnonymizeEvent
  - gdcmm::AnonymizeEvent, [108](#)
- ~Anonymizer
  - gdcmm::Anonymizer, [113](#)
- ~Attribute
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [153](#)
- ~AudioCodec
  - gdcmm::AudioCodec, [165](#)
- ~BaseCompositeMessage
  - gdcmm::network::BaseCompositeMessage, [170](#)
- ~BaseNormalizedMessage
  - gdcmm::network::BaseNormalizedMessage, [172](#)
- ~BasePDU
  - gdcmm::network::BasePDU, [174](#)
- ~BaseQuery
  - gdcmm::BaseQuery, [178](#)
- ~BaseRootQuery
  - gdcmm::BaseRootQuery, [183](#)
- ~Bitmap
  - gdcmm::Bitmap, [195](#)
- ~BitmapToBitmapFilter
  - gdcmm::BitmapToBitmapFilter, [209](#)
- ~BoxRegion
  - gdcmm::BoxRegion, [212](#)
- ~ByteSwapFilter
  - gdcmm::ByteSwapFilter, [220](#)
- ~ByteValue
  - gdcmm::ByteValue, [223](#)
- ~CAPICryptographicMessageSyntax
  - gdcmm::CAPICryptographicMessageSyntax, [232](#)
- ~CSAHeader
  - gdcmm::CSAHeader, [286](#)
- ~Coder
  - gdcmm::Coder, [248](#)
- ~Command
  - gdcmm::Command, [256](#)
- ~CommandDataSet
  - gdcmm::CommandDataSet, [258](#)
- ~CryptoFactory
  - gdcmm::CryptoFactory, [271](#)
- ~CryptographicMessageSyntax
  - gdcmm::CryptographicMessageSyntax, [273](#)
- ~Curve
  - gdcmm::Curve, [302](#)
- ~DICOMDIRGenerator
  - gdcmm::DICOMDIRGenerator, [354](#)
- ~DataEvent
  - gdcmm::DataEvent, [323](#)
- ~DataSetEvent
  - gdcmm::DataSetEvent, [339](#)
- ~Decoder
  - gdcmm::Decoder, [342](#)
- ~Defs
  - gdcmm::Defs, [346](#)
- ~DeltaEncodingCodec
  - gdcmm::DeltaEncodingCodec, [350](#)
- ~DictConverter
  - gdcmm::DictConverter, [362](#)
- ~DictPrinter
  - gdcmm::DictPrinter, [371](#)
- ~Dicts
  - gdcmm::Dicts, [374](#)
- ~DirectionCosines
  - gdcmm::DirectionCosines, [379](#)
- ~Directory
  - gdcmm::Directory, [383](#)
- ~Dumper
  - gdcmm::Dumper, [391](#)
- ~Element
  - gdcmm::Element< TVR, VM::VM1\_n >, [399](#)
- ~EmptyMaskGenerator
  - gdcmm::EmptyMaskGenerator, [418](#)
- ~Event
  - gdcmm::Event, [428](#)
- ~Exception
  - gdcmm::Exception, [431](#)
- ~File
  - gdcmm::File, [441](#)
- ~FileAnonymizer
  - gdcmm::FileAnonymizer, [446](#)
- ~FileChangeTransferSyntax
  - gdcmm::FileChangeTransferSyntax, [450](#)
- ~FileDecompressLookupTable
  - gdcmm::FileDecompressLookupTable, [453](#)
- ~FileDerivation
  - gdcmm::FileDerivation, [456](#)
- ~FileExplicitFilter

- gdcm::FileExplicitFilter, [460](#)
- ~FileMetaInformation
  - gdcm::FileMetaInformation, [465](#)
- ~FileNameEvent
  - gdcm::FileNameEvent, [477](#)
- ~FileStreamer
  - gdcm::FileStreamer, [487](#)
- ~FilenameGenerator
  - gdcm::FilenameGenerator, [480](#)
- ~Global
  - gdcm::Global, [503](#)
- ~GroupDict
  - gdcm::GroupDict, [506](#)
- ~IconImageFilter
  - gdcm::IconImageFilter, [509](#)
- ~IconImageGenerator
  - gdcm::IconImageGenerator, [513](#)
- ~Image
  - gdcm::Image, [518](#)
- ~ImageApplyLookupTable
  - gdcm::ImageApplyLookupTable, [525](#)
- ~ImageChangePhotometricInterpretation
  - gdcm::ImageChangePhotometricInterpretation, [528](#)
- ~ImageChangePlanarConfiguration
  - gdcm::ImageChangePlanarConfiguration, [532](#)
- ~ImageChangeTransferSyntax
  - gdcm::ImageChangeTransferSyntax, [536](#)
- ~ImageCodec
  - gdcm::ImageCodec, [542](#)
- ~ImageConverter
  - gdcm::ImageConverter, [553](#)
- ~ImageFragmentSplitter
  - gdcm::ImageFragmentSplitter, [556](#)
- ~ImageReader
  - gdcm::ImageReader, [566](#)
- ~ImageRegionReader
  - gdcm::ImageRegionReader, [570](#)
- ~ImageToImageFilter
  - gdcm::ImageToImageFilter, [574](#)
- ~ImageWriter
  - gdcm::ImageWriter, [577](#)
- ~JPEG12Codec
  - gdcm::JPEG12Codec, [608](#)
- ~JPEG16Codec
  - gdcm::JPEG16Codec, [611](#)
- ~JPEG2000Codec
  - gdcm::JPEG2000Codec, [615](#)
- ~JPEG8Codec
  - gdcm::JPEG8Codec, [621](#)
- ~JPEGCodec
  - gdcm::JPEGCodec, [625](#)
- ~JPEGLSCodec
  - gdcm::JPEGLSCodec, [634](#)
- ~JSON
  - gdcm::JSON, [639](#)
- ~KAKADUCodec
  - gdcm::KAKADUCodec, [642](#)
- ~LookupTable
  - gdcm::LookupTable, [650](#)
- ~MemberCommand
  - gdcm::MemberCommand, [678](#)
- ~MeshPrimitive
  - gdcm::MeshPrimitive, [683](#)
- ~ModuleEntry
  - gdcm::ModuleEntry, [698](#)
- ~MrProtocol
  - gdcm::MrProtocol, [710](#)
- ~Object
  - gdcm::Object, [740](#)
- ~OpenSSLCryptographicMessageSyntax
  - gdcm::OpenSSLCryptographicMessageSyntax, [745](#)
- ~OpenSSLP7CryptographicMessageSyntax
  - gdcm::OpenSSLP7CryptographicMessageSyntax, [750](#)
- ~Orientation
  - gdcm::Orientation, [753](#)
- ~Overlay
  - gdcm::Overlay, [759](#)
- ~PDBHeader
  - gdcm::PDBHeader, [780](#)
- ~PDFCodec
  - gdcm::PDFCodec, [784](#)
- ~PGXCodec
  - gdcm::PGXCodec, [793](#)
- ~PNMCodec
  - gdcm::PNMCodec, [826](#)
- ~PVRGCodec
  - gdcm::PVRGCodec, [867](#)
- ~ParseException
  - gdcm::ParseException, [767](#)
- ~Parser
  - gdcm::Parser, [770](#)
- ~Pixmap
  - gdcm::Pixmap, [809](#)
- ~PixmapReader
  - gdcm::PixmapReader, [815](#)
- ~PixmapToPixmapFilter
  - gdcm::PixmapToPixmapFilter, [819](#)
- ~PixmapWriter
  - gdcm::PixmapWriter, [822](#)
- ~Preamble
  - gdcm::Preamble, [830](#)
- ~Printer
  - gdcm::Printer, [853](#)
- ~PrivateDict
  - gdcm::PrivateDict, [857](#)
- ~ProgressEvent
  - gdcm::ProgressEvent, [864](#)

- ~PythonFilter
  - gdcm::PythonFilter, [870](#)
- ~QueryBase
  - gdcm::QueryBase, [872](#)
- ~RAWCodec
  - gdcm::RAWCodec, [887](#)
- ~RLECodec
  - gdcm::RLECodec, [908](#)
- ~Reader
  - gdcm::Reader, [893](#)
- ~Region
  - gdcm::Region, [900](#)
- ~Rescaler
  - gdcm::Rescaler, [903](#)
- ~SHA1
  - gdcm::SHA1, [979](#)
- ~Scanner
  - gdcm::Scanner, [919](#)
- ~Segment
  - gdcm::Segment, [928](#)
- ~SegmentReader
  - gdcm::SegmentReader, [940](#)
- ~SegmentWriter
  - gdcm::SegmentWriter, [943](#)
- ~SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [936](#)
- ~SerieHelper
  - gdcm::SerieHelper, [965](#)
- ~ServiceClassUser
  - gdcm::ServiceClassUser, [972](#)
- ~SimpleMemberCommand
  - gdcm::SimpleMemberCommand, [982](#)
- ~SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, [985](#)
- ~SmartPointer
  - gdcm::SmartPointer, [992](#)
- ~Sorter
  - gdcm::Sorter, [1000](#)
- ~Spacing
  - gdcm::Spacing, [1005](#)
- ~SplitMosaicFilter
  - gdcm::SplitMosaicFilter, [1007](#)
- ~StreamImageReader
  - gdcm::StreamImageReader, [1012](#)
- ~StreamImageWriter
  - gdcm::StreamImageWriter, [1018](#)
- ~StrictScanner
  - gdcm::StrictScanner, [1026](#)
- ~StringFilter
  - gdcm::StringFilter, [1038](#)
- ~Subject
  - gdcm::Subject, [1043](#)
- ~Surface
  - gdcm::Surface, [1049](#)
- ~SurfaceReader
  - gdcm::SurfaceReader, [1065](#)
- ~SurfaceWriter
  - gdcm::SurfaceWriter, [1068](#)
- ~Table
  - gdcm::Table, [1082](#)
- ~TableEntry
  - gdcm::TableEntry, [1084](#)
- ~TableReader
  - gdcm::TableReader, [1085](#)
- ~TableRow
  - gdcm::network::TableRow, [1089](#)
- ~TagPath
  - gdcm::TagPath, [1102](#)
- ~Testing
  - gdcm::Testing, [1106](#)
- ~Trace
  - gdcm::Trace, [1113](#)
- ~Transition
  - gdcm::network::Transition, [1127](#)
- ~ULAction
  - gdcm::network::ULAction, [1162](#)
- ~ULBasicCallback
  - gdcm::network::ULBasicCallback, [1202](#)
- ~ULConnection
  - gdcm::network::ULConnection, [1204](#)
- ~ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1209](#)
- ~ULConnectionManager
  - gdcm::network::ULConnectionManager, [1215](#)
- ~ULEvent
  - gdcm::network::ULEvent, [1222](#)
- ~ULWritingCallback
  - gdcm::network::ULWritingCallback, [1226](#)
- ~UserInformation
  - gdcm::network::UserInformation, [1238](#)
- ~Validate
  - gdcm::Validate, [1242](#)
- ~Value
  - gdcm::Value, [1245](#)
- ~Version
  - gdcm::Version, [1249](#)
- ~Writer
  - gdcm::Writer, [1396](#)
- ~XMLDictReader
  - gdcm::XMLDictReader, [1401](#)
- ~XMLPrinter
  - gdcm::XMLPrinter, [1404](#)
- ~XMLPrivateDictReader
  - gdcm::XMLPrivateDictReader, [1408](#)
- ~vtkGDCMImageReader
  - vtkGDCMImageReader, [1275](#)
- ~vtkGDCMImageReader2
  - vtkGDCMImageReader2, [1290](#)

- ~vtkGDCMImageWriter
  - vtkGDCMImageWriter, [1305](#)
- ~vtkGDCMMedicalImageProperties
  - vtkGDCMMedicalImageProperties, [1313](#)
- ~vtkGDCMPolyDataReader
  - vtkGDCMPolyDataReader, [1317](#)
- ~vtkGDCMPolyDataWriter
  - vtkGDCMPolyDataWriter, [1322](#)
- ~vtkGDCMTesting
  - vtkGDCMTesting, [1327](#)
- ~vtkGDCMThreadedImageReader
  - vtkGDCMThreadedImageReader, [1331](#)
- ~vtkGDCMThreadedImageReader2
  - vtkGDCMThreadedImageReader2, [1335](#)
- ~vtkImageColorViewer
  - vtkImageColorViewer, [1345](#)
- ~vtkImageMapToColors16
  - vtkImageMapToColors16, [1358](#)
- ~vtkImageMapToWindowLevelColors2
  - vtkImageMapToWindowLevelColors2, [1364](#)
- ~vtkImagePlanarComponentsToComponents
  - vtkImagePlanarComponentsToComponents, [1368](#)
- ~vtkImageRGBToYBR
  - vtkImageRGBToYBR, [1370](#)
- ~vtkImageYBRToRGB
  - vtkImageYBRToRGB, [1373](#)
- ~vtkLookupTable16
  - vtkLookupTable16, [1376](#)
- ~vtkRTStructSetProperties
  - vtkRTStructSetProperties, [1380](#)
- AAAbortPDU
  - gdcm::network::AAAbortPDU, [88](#)
- AAAssociateACPDU
  - gdcm::network::AAAssociateACPDU, [91](#)
  - gdcm::network::AAAssociateRQPDU, [103](#)
- AAAssociateRJPDU
  - gdcm::network::AAAssociateRJPDU, [95](#)
- AAAssociateRQPDU
  - gdcm::network::AAAssociateACPDU, [94](#)
  - gdcm::network::AAAssociateRQPDU, [99](#)
- AECComp
  - gdcm, [58](#)
- ALGOType
  - gdcm::Segment, [927](#)
- ARTIMTimer
  - gdcm::network::ARTIMTimer, [129](#)
- AReleaseRPPDU
  - gdcm::network::AReleaseRPPDU, [125](#)
- AReleaseRQPDU
  - gdcm::network::AReleaseRQPDU, [127](#)
- ASComp
  - gdcm, [58](#)
- ASN1
  - gdcm::ASN1, [131](#)
- AbstractSyntax
  - gdcm::PresentationContext, [836](#)
  - gdcm::network::AbstractSyntax, [105](#)
- ActiveComponent
  - vtkImageMapToColors16, [1362](#)
- Add
  - gdcm::GroupDict, [507](#)
- AddAcceptedPresentationContext
  - gdcm::network::ULConnection, [1204](#)
- AddCSAHeaderDictEntry
  - gdcm::CSAHeaderDict, [291](#)
- AddContourReferencedFrameOfReference
  - vtkRTStructSetProperties, [1380](#)
- AddDerivationDescription
  - gdcm::FileDerivation, [456](#)
- AddDictEntry
  - gdcm::Dict, [358](#)
  - gdcm::PrivateDict, [857](#)
- AddFile
  - gdcm::FileSet, [484](#)
  - gdcm::SerieHelper, [965](#)
- AddFileName
  - gdcm::SerieHelper, [965](#)
- AddFragment
  - gdcm::SequenceOfFragments, [948](#)
- AddFromFile
  - gdcm::PresentationContextGenerator, [841](#)
- AddGroupLength
  - gdcm::DictConverter, [362](#)
- AddIODEntry
  - gdcm::IOD, [588](#)
- AddIOD
  - gdcm::IODs, [594](#)
- AddImageDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddInput
  - vtkImageColorViewer, [1345](#)
- AddInputConnection
  - vtkImageColorViewer, [1345](#)
- AddItem
  - gdcm::SequenceOfItems, [957](#)
- AddMacro
  - gdcm::Macros, [661](#)
  - gdcm::Module, [694](#)
- AddMacroEntry
  - gdcm::Macro, [658](#)
- AddModule
  - gdcm::Modules, [702](#)
- AddModuleEntry
  - gdcm::Module, [694](#)
  - gdcm::NestedModuleEntries, [723](#)
- AddNewUndefinedLengthItem
  - gdcm::SequenceOfItems, [957](#)

- AddObserver
  - gdcm::Subject, [1043](#), [1044](#)
- AddPatientDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddPresentationContext
  - gdcm::PresentationContextGenerator, [841](#)
  - gdcm::network::AAssociateRQPDU, [99](#)
- AddPresentationContextAC
  - gdcm::network::AAssociateACPDU, [92](#)
- AddPresentationDataValue
  - gdcm::network::PDataTFPDU, [774](#)
- AddPrimitiveData
  - gdcm::MeshPrimitive, [683](#)
- AddPrivateTag
  - gdcm::Scanner, [919](#)
  - gdcm::StrictScanner, [1026](#)
- AddPurposeOfReferenceCodeSequence
  - gdcm::FileDerivation, [456](#)
- AddQueryDataSet
  - gdcm::BaseQuery, [178](#)
- AddReference
  - gdcm::FileDerivation, [456](#)
- AddReferencedFrameOfReference
  - vtkRTStructSetProperties, [1380](#)
- AddRestriction
  - gdcm::SerieHelper, [965](#)
- AddRoleSelectionSub
  - gdcm::network::UserInformation, [1238](#)
- AddSOPClassExtendedNegociationSub
  - gdcm::network::UserInformation, [1238](#)
- AddSegment
  - gdcm::SegmentWriter, [944](#)
- AddSelect
  - gdcm::Sorter, [1000](#)
- AddSeriesDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddSkipTag
  - gdcm::Scanner, [919](#)
  - gdcm::StrictScanner, [1027](#)
- AddSourceImageSequence
  - gdcm::FileDerivation, [456](#)
- AddStructureSetROIObservation
  - vtkRTStructSetProperties, [1381](#)
- AddStructureSetROI
  - vtkRTStructSetProperties, [1381](#)
- AddStudyDirectoryRecord
  - gdcm::DICOMDIRGenerator, [354](#)
- AddSurface
  - gdcm::Segment, [928](#)
- AddTag
  - gdcm::Scanner, [919](#)
  - gdcm::StrictScanner, [1027](#)
- AddTransferSyntax
  - gdcm::PresentationContext, [835](#)
  - gdcm::network::PresentationContextRQ, [845](#)
- AffectedSOPClassUID
  - gdcm::network::CEchoRQ, [236](#)
- Allocate
  - gdcm::LookupTable, [650](#)
- AnatomicRegion
  - gdcm::Segment, [933](#)
- AnatomicRegionModifiers
  - gdcm::Segment, [933](#)
- AnonymizeEvent
  - gdcm::AnonymizeEvent, [108](#), [109](#)
- Anonymizer
  - gdcm::Anonymizer, [113](#)
- Append
  - gdcm::ByteValue, [223](#)
  - gdcm::Global, [503](#)
- AppendFrameEncode
  - gdcm::ImageCodec, [542](#)
  - gdcm::JPEG2000Codec, [615](#)
  - gdcm::JPEGCodec, [625](#)
  - gdcm::JPEGLSCodec, [634](#)
  - gdcm::RLECodec, [908](#)
- AppendImplementationClassUID
  - gdcm::FileMetaInformation, [465](#)
- AppendRowEncode
  - gdcm::ImageCodec, [542](#)
  - gdcm::JPEG2000Codec, [615](#)
  - gdcm::JPEGCodec, [626](#)
  - gdcm::JPEGLSCodec, [634](#)
  - gdcm::RLECodec, [908](#)
- AppendToDataElement
  - gdcm::FileStreamer, [487](#)
- AppendToGroupDataElement
  - gdcm::FileStreamer, [487](#)
- ApplicationContext
  - gdcm::network::ApplicationContext, [120](#)
- Apply
  - gdcm::ImageApplyLookupTable, [525](#)
- ApplyInverseVideo
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1299](#)
- ApplyLookupTable
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1299](#)
- ApplyPlanarConfiguration
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1299](#)
- ApplyShiftScale
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1299](#)
- ApplyYBRToRGB
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1299](#)
- AreOverlaysInPixelData

- gdcm::Bitmap, [195](#)
- gdcm::Pixmap, [809](#)
- Area
  - gdcm::BoxRegion, [212](#)
  - gdcm::Region, [900](#)
- ArrayIncludeMacrosType
  - gdcm::Macro, [657](#)
  - gdcm::Module, [694](#)
- ArrayType
  - gdcm::Attribute, [136](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [143](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [153](#)
- AsynchronousOperationsWindowSub
  - gdcm::network::AsynchronousOperationsWindow↔Sub, [132](#)
- Attribute
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [153](#)
  - gdcm::terminal, [85](#)
- AudioCodec
  - gdcm::AudioCodec, [165](#)
- AutoPixelMinMax
  - gdcm::IconImageGenerator, [513](#)
- BALCPPProtect
  - gdcm::Anonymizer, [113](#)
- BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER
  - gdcm, [58](#)
- backslash
  - gdcm, [63](#)
- BaseQuery
  - gdcm::BaseQuery, [178](#)
- BaseRootQuery
  - gdcm::BaseRootQuery, [183](#)
- BasicApplicationLevelConfidentialityProfile
  - gdcm::Anonymizer, [113](#)
- BasicCodedEntry
  - gdcm::SegmentHelper::BasicCodedEntry, [188](#)
- BasicCodedEntryVector
  - gdcm::Segment, [927](#)
- BasicOffsetTable
  - gdcm::BasicOffsetTable, [191](#)
- Begin
  - gdcm::CSAHeaderDict, [291](#)
  - gdcm::DataSet, [328](#)
  - gdcm::Dict, [358](#)
  - gdcm::IODs, [594](#)
  - gdcm::Scanner, [920](#)
  - gdcm::SequenceOfFragments, [949](#)
  - gdcm::SequenceOfItems, [957](#)
  - gdcm::StrictScanner, [1027](#)
- BitSample
  - gdcm::JPEGCodec, [631](#)
  - gdcm::LookupTable, [655](#)
- Bitmap
  - gdcm::Bitmap, [195](#)
  - gdcm::JPEG2000Codec, [619](#)
  - gdcm::PixelFormat, [807](#)
- BitmapToBitmapFilter
  - gdcm::BitmapToBitmapFilter, [209](#)
- BoundingBox
  - gdcm::BoxRegion, [212](#)
- BoxRegion
  - gdcm::BoxRegion, [212](#)
- BreakConnection
  - gdcm::network::ULConnectionManager, [1216](#)
- BreakConnectionNow
  - gdcm::network::ULConnectionManager, [1216](#)
- Build
  - vtkLookupTable16, [1376](#)
- ByteBuffer
  - gdcm::ByteBuffer, [216](#)
- ByteSwap
  - gdcm::ByteSwapFilter, [220](#)
- ByteSwapFilter
  - gdcm::ByteSwapFilter, [219](#)
- ByteValue
  - gdcm::ByteValue, [223](#)
- bytes
  - gdcm::Tag, [1101](#)
- CAPICryptoFactory
  - gdcm::CAPICryptoFactory, [230](#)
- CAPICryptographicMessageSyntax
  - gdcm::CAPICryptographicMessageSyntax, [232](#)
- CEcho
  - gdcm::CompositeNetworkFunctions, [263](#)
- CFind
  - gdcm::CompositeNetworkFunctions, [263](#)
- cMaxEventID
  - gdcm::network, [83](#)
- cMaxStateID
  - gdcm::network, [83](#)
- CMove
  - gdcm::CompositeNetworkFunctions, [264](#)
- CSAElement
  - gdcm::CSAElement, [277](#)
- CSAHeader
  - gdcm::CSAHeader, [286](#)
  - gdcm::DataSet, [337](#)
- CSAHeaderDict
  - gdcm::CSAHeaderDict, [291](#)
- CSAHeaderDictEntry
  - gdcm::CSAHeaderDictEntry, [294](#)
- CSAHeaderType
  - gdcm::CSAHeader, [285](#)



- CSComp
  - gdcm, [58](#)
- CSD
  - gdcm::SegmentHelper::BasicCodedEntry, [189](#)
- CStore
  - gdcm::CompositeNetworkFunctions, [265](#)
- CSV
  - gdcm::SegmentHelper::BasicCodedEntry, [189](#)
- CanCode
  - gdcm::AudioCodec, [166](#)
  - gdcm::Coder, [248](#)
  - gdcm::ImageCodec, [543](#)
  - gdcm::JPEG2000Codec, [615](#)
  - gdcm::JPEGCodec, [626](#)
  - gdcm::JPEGLSCodec, [634](#)
  - gdcm::KAKADUCodec, [642](#)
  - gdcm::PDFCodec, [784](#)
  - gdcm::PGXCodec, [793](#)
  - gdcm::PNMCodec, [826](#)
  - gdcm::PVRGCodec, [867](#)
  - gdcm::RAWCodec, [887](#)
  - gdcm::RLECodec, [909](#)
- CanDecode
  - gdcm::AudioCodec, [166](#)
  - gdcm::Decoder, [343](#)
  - gdcm::DeltaEncodingCodec, [350](#)
  - gdcm::ImageCodec, [543](#)
  - gdcm::JPEG2000Codec, [616](#)
  - gdcm::JPEGCodec, [626](#)
  - gdcm::JPEGLSCodec, [634](#)
  - gdcm::KAKADUCodec, [642](#)
  - gdcm::PDFCodec, [784](#)
  - gdcm::PGXCodec, [793](#)
  - gdcm::PNMCodec, [827](#)
  - gdcm::PVRGCodec, [868](#)
  - gdcm::RAWCodec, [888](#)
  - gdcm::RLECodec, [909](#)
- CanDisplay
  - gdcm::VR, [1263](#)
- CanEmptyTag
  - gdcm::Anonymizer, [114](#)
- CanRead
  - gdcm::Reader, [893](#)
- CanReadFile
  - vtkGDCMImageReader, [1276](#)
  - vtkGDCMImageReader2, [1291](#)
- CanReadImage
  - gdcm::StreamImageReader, [1012](#)
- CanStoreLossy
  - gdcm::TransferSyntax, [1121](#)
- CanWriteFile
  - gdcm::StreamImageWriter, [1018](#)
- Change
  - gdcm::FileChangeTransferSyntax, [450](#)
  - gdcm::FileDecompressLookupTable, [453](#)
  - gdcm::FileExplicitFilter, [460](#)
  - gdcm::ImageChangePhotometricInterpretation, [528](#)
  - gdcm::ImageChangePlanarConfiguration, [532](#)
  - gdcm::ImageChangeTransferSyntax, [537](#)
- ChangeFMI
  - gdcm::FileExplicitFilter, [460](#)
- ChangeMonochrome
  - gdcm::ImageChangePhotometricInterpretation, [529](#)
- ChangeRGB2YBR
  - gdcm::ImageChangePhotometricInterpretation, [529](#)
- ChangeYBR2RGB
  - gdcm::ImageChangePhotometricInterpretation, [529](#)
- CharacterDataHandler
  - gdcm::TableReader, [1086](#)
  - gdcm::XMLDictReader, [1401](#)
  - gdcm::XMLPrivateDictReader, [1408](#)
- CheckDataElement
  - gdcm::FileStreamer, [488](#)
- CheckEvent
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::DataEvent, [324](#)
  - gdcm::DataSetEvent, [340](#)
  - gdcm::Event, [428](#)
  - gdcm::FileNameEvent, [477](#)
  - gdcm::ProgressEvent, [864](#)
- CheckFileMetaInformationOff
  - gdcm::Writer, [1396](#)
- CheckFileMetaInformationOn
  - gdcm::Writer, [1396](#)
- CheckTemplateFileName
  - gdcm::FileStreamer, [488](#)
- CipherTypes
  - gdcm::CryptographicMessageSyntax, [272](#)
- Clamp
  - gdcm, [63](#)
- CleanupUnusedBits
  - gdcm::ImageCodec, [543](#)
- Clear
  - gdcm::Bitmap, [196](#)
  - gdcm::ByteValue, [224](#)
  - gdcm::DataElement, [310](#)
  - gdcm::DataSet, [329](#)
  - gdcm::IODs, [595](#)
  - gdcm::IOD, [588](#)
  - gdcm::Item, [603](#)
  - gdcm::LookupTable, [651](#)
  - gdcm::Macro, [658](#)
  - gdcm::Macros, [661](#)
  - gdcm::Module, [694](#)
  - gdcm::Modules, [702](#)
  - gdcm::Preamble, [830](#)
  - gdcm::SequenceOfFragments, [949](#)
  - gdcm::SequenceOfItems, [957](#)

- gdcM::SerieHelper, 966
- gdcM::Value, 1245
- vtkGDCMMedicalImageProperties, 1313
- vtkRTStructSetProperties, 1381
- ClearInternalUIDs
  - gdcM::Anonymizer, 114
- ClearSkipTags
  - gdcM::Scanner, 920
  - gdcM::StrictScanner, 1027
- ClearTags
  - gdcM::Scanner, 920
  - gdcM::StrictScanner, 1027
- Clone
  - gdcM::BoxRegion, 213
  - gdcM::ImageCodec, 543
  - gdcM::JPEG2000Codec, 616
  - gdcM::JPEGCodec, 626
  - gdcM::JPEGLSCoDec, 635
  - gdcM::KAKADUCoDec, 643
  - gdcM::PGXCoDec, 794
  - gdcM::PNMCoDec, 827
  - gdcM::PVRGCoDec, 868
  - gdcM::RAWCoDec, 888
  - gdcM::RLECoDec, 909
  - gdcM::Region, 900
- CM
  - gdcM::SegmentHelper::BasicCodedEntry, 189
- Code
  - gdcM::Coder, 248
  - gdcM::JPEG2000Codec, 616
  - gdcM::JPEGCodec, 627
  - gdcM::JPEGLSCoDec, 635
  - gdcM::JSON, 639
  - gdcM::KAKADUCoDec, 643
  - gdcM::PVRGCoDec, 868
  - gdcM::RAWCoDec, 888
  - gdcM::RLECoDec, 909
- CodeMeaning
  - gdcM::RealWorldValueMappingContent, 898
- CodeString
  - gdcM::CodeString, 252
- CodeValue
  - gdcM::RealWorldValueMappingContent, 898
- Color
  - gdcM::terminal, 85
- ColorArray
  - gdcM::SurfaceHelper, 1061
- Command
  - gdcM::Command, 255
- CommandDataSet
  - gdcM::CommandDataSet, 258
- CommandTypes
  - gdcM::network::DIMSE, 377
- CompOperators
  - gdcM, 60
- Compatible
  - gdcM::VM, 1258
  - gdcM::VR, 1264
- Component
  - gdcM::PersonName, 791
- CompressionTypes
  - vtkGDCMImageWriter, 1304
- Compute
  - gdcM::MD5, 665
  - gdcM::SHA1, 979
- ComputeBoundingBox
  - gdcM::BoxRegion, 213
  - gdcM::Region, 900
- ComputeBufferLength
  - gdcM::ImageRegionReader, 570
- ComputeByteLength
  - gdcM::SequenceOfFragments, 949
- ComputeDataElement
  - gdcM::DataSet, 329
- ComputeDataSetMediaStorageSOPClass
  - gdcM::FileMetaInformation, 465
- ComputeDataSetTransferSyntax
  - gdcM::FileMetaInformation, 465
- ComputeDistAlongNormal
  - gdcM::DirectionCosines, 379
- ComputeFile
  - gdcM::MD5, 665
  - gdcM::SHA1, 979
- ComputeFileMD5
  - gdcM::Testing, 1106
- ComputeGroupLength
  - gdcM::DataSet, 329
- ComputeInterceptSlopePixelType
  - gdcM::Rescaler, 903
- ComputeLength
  - gdcM::ByteValue, 224
  - gdcM::Fragment, 500
  - gdcM::SequenceOfFragments, 949
  - gdcM::SequenceOfItems, 958
- ComputeLossyFlag
  - gdcM::Bitmap, 196
- ComputeMD5
  - gdcM::Testing, 1106
- ComputeMOSAICDimensions
  - gdcM::SplitMosaicFilter, 1007
- ComputeMOSAICSliceNormal
  - gdcM::SplitMosaicFilter, 1007
- ComputeMOSAICSlicePosition
  - gdcM::SplitMosaicFilter, 1007
- ComputeMediaStorageFromModality
  - gdcM::ImageHelper, 558
- ComputeNumberOfSurfaces
  - gdcM::SurfaceWriter, 1068

- ComputeOffsetTable
  - gdcm::JPEGCodec, [627](#)
- ComputePixelAspectRatioFromPixelSpacing
  - gdcm::Spacing, [1005](#)
- ComputePixelTypeFromMinMax
  - gdcm::Rescaler, [903](#)
- ComputeSpacingFromImagePositionPatient
  - gdcm::ImageHelper, [558](#)
- ComputeTargetMediaStorage
  - gdcm::ImageWriter, [577](#)
- ComputeVR
  - gdcm::DataSetHelper, [341](#)
- ComputeZSpacing
  - gdcm::IPPSorter, [600](#)
- ConcatenatePDVBlobs
  - gdcm::network::PresentationDataValue, [848](#)
- ConcatenatePDVBlobsAsExplicit
  - gdcm::network::PresentationDataValue, [848](#)
- const
  - gdcm::SOPClassUIDToIOD, [996](#)
- const\_iterator
  - gdcm::CodeString, [250](#)
  - gdcm::LO, [645](#)
  - gdcm::String, [1034](#)
- const\_reference
  - gdcm::CodeString, [251](#)
  - gdcm::LO, [645](#)
  - gdcm::String, [1034](#)
- const\_reverse\_iterator
  - gdcm::CodeString, [251](#)
  - gdcm::LO, [645](#)
  - gdcm::String, [1034](#)
- ConstCharWrapper
  - gdcm::ConstCharWrapper, [266](#)
- ConstIterator
  - gdcm::CSAHeaderDict, [290](#)
  - gdcm::DataSet, [328](#)
  - gdcm::Dict, [357](#)
  - gdcm::Scanner, [918](#)
  - gdcm::SequenceOfFragments, [948](#)
  - gdcm::SequenceOfItems, [956](#)
  - gdcm::StrictScanner, [1025](#)
- Construct
  - gdcm::BaseRootQuery, [184](#)
- ConstructAbortPDU
  - gdcm::network::PDUFactory, [786](#)
- ConstructCEchoRQ
  - gdcm::network::CompositeMessageFactory, [260](#)
- ConstructCFindRQ
  - gdcm::network::CompositeMessageFactory, [260](#)
- ConstructCMoveRQ
  - gdcm::network::CompositeMessageFactory, [260](#)
- ConstructCStoreRSP
  - gdcm::network::CompositeMessageFactory, [261](#)
- ConstructCStoreRQ
  - gdcm::network::CompositeMessageFactory, [261](#)
- ConstructFromString
  - gdcm::TagPath, [1102](#)
- ConstructFromTagList
  - gdcm::TagPath, [1103](#)
- ConstructNAction
  - gdcm::network::NormalizedMessageFactory, [731](#)
- ConstructNCreate
  - gdcm::network::NormalizedMessageFactory, [731](#)
- ConstructNDelete
  - gdcm::network::NormalizedMessageFactory, [731](#)
- ConstructNEventReport
  - gdcm::network::NormalizedMessageFactory, [732](#)
- ConstructNGet
  - gdcm::network::NormalizedMessageFactory, [732](#)
- ConstructNSet
  - gdcm::network::NormalizedMessageFactory, [732](#)
- ConstructPDVByDataSet
  - gdcm::network::CEchoRSP, [237](#)
  - gdcm::network::CFindCancelRQ, [239](#)
  - gdcm::network::CFindRSP, [242](#)
  - gdcm::network::CMoveCancelRq, [243](#)
  - gdcm::network::CMoveRSP, [246](#)
  - gdcm::network::NActionRSP, [714](#)
  - gdcm::network::NCreateRSP, [717](#)
  - gdcm::network::NDeleteRSP, [720](#)
  - gdcm::network::NEventReportRSP, [726](#)
  - gdcm::network::NGetRSP, [729](#)
  - gdcm::network::NSetRSP, [738](#)
- ConstructPDU
  - gdcm::network::PDUFactory, [786](#)
- ConstructPDV
  - gdcm::network::BaseCompositeMessage, [170](#)
  - gdcm::network::BaseNormalizedMessage, [172](#)
  - gdcm::network::CEchoRQ, [235](#)
  - gdcm::network::CFindRQ, [240](#)
  - gdcm::network::CMoveRQ, [244](#)
  - gdcm::network::CStoreRSP, [299](#)
  - gdcm::network::CStoreRQ, [298](#)
  - gdcm::network::NActionRQ, [713](#)
  - gdcm::network::NCreateRQ, [716](#)
  - gdcm::network::NDeleteRQ, [719](#)
  - gdcm::network::NEventReportRQ, [725](#)
  - gdcm::network::NGetRQ, [728](#)
  - gdcm::network::NSetRQ, [736](#)
- ConstructQuery
  - gdcm::CompositeNetworkFunctions, [264](#), [265](#)
  - gdcm::NormalizedNetworkFunctions, [733](#)
- ConstructReleasePDU
  - gdcm::network::PDUFactory, [786](#)
- ConstructorType
  - gdcm::Dicts, [373](#)
- Convert

- gdcm::DictConverter, [362](#)
- gdcm::ImageConverter, [553](#)
- ConvertRGBToPaletteColor
  - gdcm::IconImageGenerator, [513](#)
- ConvertToCXX
  - gdcm::DictConverter, [362](#)
- ConvertToXML
  - gdcm::DictConverter, [362](#)
- Create
  - gdcm::Preamble, [830](#)
- CreateCEchoPDU
  - gdcm::network::PDUFactory, [786](#)
- CreateCFindPDU
  - gdcm::network::PDUFactory, [787](#)
- CreateCMSProvider
  - gdcm::CAPICryptoFactory, [230](#)
  - gdcm::CryptoFactory, [271](#)
  - gdcm::OpenSSLCryptoFactory, [743](#)
  - gdcm::OpenSSL7CryptoFactory, [748](#)
- CreateCMovePDU
  - gdcm::network::PDUFactory, [787](#)
- CreateCStoreRQPDU
  - gdcm::network::PDUFactory, [787](#)
- CreateCStoreRSPPDU
  - gdcm::network::PDUFactory, [787](#)
- CreateDefaultUniqueSeriesIdentifier
  - gdcm::SerieHelper, [966](#)
- CreateNActionPDU
  - gdcm::network::PDUFactory, [787](#)
- CreateNCreatePDU
  - gdcm::network::PDUFactory, [787](#)
- CreateNDeletePDU
  - gdcm::network::PDUFactory, [788](#)
- CreateNEventReportPDU
  - gdcm::network::PDUFactory, [788](#)
- CreateNGetPDU
  - gdcm::network::PDUFactory, [788](#)
- CreateNSetPDU
  - gdcm::network::PDUFactory, [788](#)
- CreateUniqueSeriesIdentifier
  - gdcm::SerieHelper, [966](#)
- Cross
  - gdcm::DirectionCosines, [379](#)
- CrossDot
  - gdcm::DirectionCosines, [379](#)
- CryptoFactory
  - gdcm::CryptoFactory, [271](#)
- CryptoLib
  - gdcm::CryptoFactory, [270](#)
- CryptographicMessageSyntax
  - gdcm::CryptographicMessageSyntax, [273](#)
- Curve
  - gdcm::Curve, [301](#), [302](#)
  - vtkGDCMImageReader, [1285](#)
- vtkGDCMImageReader2, [1299](#)
- Curves
  - gdcm::Pixmap, [812](#)
- CV
  - gdcm::SegmentHelper::BasicCodedEntry, [189](#)
- DAComp
  - gdcm, [58](#)
- dCor
  - gdcm::MrProtocol::Vector3, [1248](#)
- DICOMDIRGenerator
  - gdcm::DICOMDIRGenerator, [354](#)
- DICOMDIR
  - gdcm::DICOMDIR, [351](#), [352](#)
- dSag
  - gdcm::MrProtocol::Vector3, [1248](#)
- DTComp
  - gdcm, [58](#)
- dTra
  - gdcm::MrProtocol::Vector3, [1248](#)
- DataElement
  - gdcm::DataElement, [309](#), [310](#)
  - gdcm::Value, [1246](#)
- DataElementSet
  - gdcm::DataSet, [328](#)
- DataElementType
  - gdcm::ModuleEntry, [700](#)
- DataEvent
  - gdcm::DataEvent, [323](#), [324](#)
- DataField
  - gdcm::CSAElement, [282](#)
- DataPtr
  - gdcm::CSAElement, [277](#)
- DataSetEvent
  - gdcm::DataSetEvent, [339](#), [340](#)
- DataSetHandled
  - gdcm::network::ULConnectionCallback, [1210](#)
- DataSetHandles
  - gdcm::network::ULConnectionCallback, [1210](#)
- DataSetMS
  - gdcm::FileMetaInformation, [471](#)
- DataSetTS
  - gdcm::FileMetaInformation, [471](#)
- DataWasPassed
  - vtkImageMapToColors16, [1362](#)
- DebugOff
  - gdcm::Trace, [1113](#)
- DebugOn
  - gdcm::Trace, [1113](#)
- Decode
  - gdcm::AudioCodec, [166](#)
  - gdcm::Base64, [167](#)
  - gdcm::Curve, [302](#)
  - gdcm::Decoder, [343](#)

- gdcmm::DeltaEncodingCodec, [350](#), [351](#)
- gdcmm::ImageCodec, [544](#)
- gdcmm::JPEG2000Codec, [616](#)
- gdcmm::JPEGCodec, [627](#)
- gdcmm::JPEGLSCodec, [635](#)
- gdcmm::JSON, [639](#)
- gdcmm::KAKADUCodec, [643](#)
- gdcmm::LookupTable, [651](#)
- gdcmm::PDFCodec, [785](#)
- gdcmm::PVRGCodec, [868](#)
- gdcmm::RAWCodec, [888](#)
- gdcmm::RLECodec, [909](#)
- DecodeByStreams
  - gdcmm::Decoder, [343](#)
  - gdcmm::ImageCodec, [544](#)
  - gdcmm::JPEG12Codec, [608](#)
  - gdcmm::JPEG16Codec, [611](#)
  - gdcmm::JPEG2000Codec, [617](#)
  - gdcmm::JPEG8Codec, [622](#)
  - gdcmm::JPEGCodec, [627](#)
  - gdcmm::RAWCodec, [889](#)
  - gdcmm::RLECodec, [910](#)
- DecodeBytes
  - gdcmm::RAWCodec, [889](#)
- DecodeExtent
  - gdcmm::JPEG2000Codec, [617](#)
  - gdcmm::JPEGCodec, [627](#)
  - gdcmm::JPEGLSCodec, [636](#)
  - gdcmm::RLECodec, [910](#)
- Decompress
  - gdcmm::Overlay, [759](#)
- Decrypt
  - gdcmm::CAPICryptographicMessageSyntax, [232](#)
  - gdcmm::CryptographicMessageSyntax, [273](#)
  - gdcmm::OpenSSLCryptographicMessageSyntax, [745](#)
  - gdcmm::OpenSSL7CryptographicMessageSyntax, [750](#)
- DeepCopy
  - vtkRTStructSetProperties, [1381](#)
- Default
  - gdcmm::FileMetaInformation, [465](#)
- DefinePixelExtent
  - gdcmm::StreamImageReader, [1012](#)
  - gdcmm::StreamImageWriter, [1018](#)
- DefineProperBufferLength
  - gdcmm::StreamImageReader, [1013](#)
  - gdcmm::StreamImageWriter, [1018](#)
- DefinedTerms
  - gdcmm::DefinedTerms, [344](#)
- Defs
  - gdcmm::Defs, [345](#)
- DeleteDirectory
  - gdcmm::System, [1075](#)
- DeltaEncodingCodec
  - gdcmm::DeltaEncodingCodec, [350](#)
- Derive
  - gdcmm::FileDerivation, [457](#)
- Description
  - gdcmm::ModuleEntry, [698](#)
- DescriptionField
  - gdcmm::ModuleEntry, [700](#)
- DetermineEventByPDU
  - gdcmm::network::PDUFactory, [788](#)
- Dict
  - gdcmm::Dict, [358](#)
  - gdcmm::DictEntry, [369](#)
- DictConverter
  - gdcmm::DictConverter, [362](#)
- DictEntry
  - gdcmm::DictEntry, [366](#)
- DictPrinter
  - gdcmm::DictPrinter, [371](#)
- Dicts
  - gdcmm::CSAHeaderDict, [292](#)
  - gdcmm::Dict, [360](#)
  - gdcmm::Dicts, [373](#)
  - gdcmm::PrivateDict, [859](#)
- difference\_type
  - gdcmm::CodeString, [251](#)
  - gdcmm::LO, [645](#)
  - gdcmm::String, [1034](#)
- Dimensions
  - gdcmm::Bitmap, [206](#)
  - gdcmm::ImageCodec, [551](#)
- DirCosTolerance
  - gdcmm::IPPSorter, [600](#)
- DirectionCosines
  - gdcmm::DirectionCosines, [379](#)
  - vtkGDCMImageReader, [1285](#)
  - vtkGDCMImageReader2, [1300](#)
- Directory
  - gdcmm::Directory, [383](#)
- DoByteSwap
  - gdcmm::ImageCodec, [544](#)
- DolconImage
  - gdcmm::PixmapWriter, [822](#)
- DolInvertMonochrome
  - gdcmm::ImageCodec, [544](#)
- DoOverlayCleanup
  - gdcmm::ImageCodec, [545](#)
- DoPaddedCompositePixelCode
  - gdcmm::ImageCodec, [545](#)
- DoPlanarConfiguration
  - gdcmm::ImageCodec, [545](#)
- DoSimpleCopy
  - gdcmm::ImageCodec, [545](#)
- DoYBR
  - gdcmm::ImageCodec, [545](#)

- Dot
  - gdcm::DirectionCosines, [380](#)
- DropDuplicatePositions
  - gdcm::IPPSorter, [600](#)
- Dumper
  - gdcm::Dumper, [390](#)
- ECharSet
  - gdcm, [61](#)
- EEventID
  - gdcm::network, [81](#)
- ENQueryType
  - gdcm, [61](#)
- EQueryLevel
  - gdcm, [61](#)
- EQueryType
  - gdcm, [62](#)
- ERootType
  - gdcm, [62](#)
- EStateID
  - gdcm::network, [82](#)
- elem
  - gdcm::SerieHelper::Rule, [914](#)
- Element
  - gdcm::Element< TVR, VM::VM1\_n >, [399](#)
- Empty
  - gdcm::Anonymizer, [114](#)
  - gdcm::BoxRegion, [213](#)
  - gdcm::DataElement, [310](#)
  - gdcm::FileAnonymizer, [446](#)
  - gdcm::Region, [900](#)
- EmptyMaskGenerator
  - gdcm::EmptyMaskGenerator, [418](#)
- EncapsulatedDocument
  - gdcm::EncapsulatedDocument, [420](#)
- Encode
  - gdcm::Base64, [168](#)
- EncodeBuffer
  - gdcm::JPEG12Codec, [608](#)
  - gdcm::JPEG16Codec, [612](#)
  - gdcm::JPEG8Codec, [622](#)
  - gdcm::JPEGCodec, [628](#)
- EncodeBytes
  - gdcm::System, [1075](#)
- Encrypt
  - gdcm::CAPICryptographicMessageSyntax, [232](#)
  - gdcm::CryptographicMessageSyntax, [273](#)
  - gdcm::OpenSSLCryptographicMessageSyntax, [745](#)
  - gdcm::OpenSSL7CryptographicMessageSyntax, [750](#)
- End
  - gdcm::CSAHeaderDict, [291](#)
  - gdcm::DataSet, [329](#)
  - gdcm::Dict, [358](#)
  - gdcm::IODs, [595](#)
  - gdcm::Scanner, [920](#)
  - gdcm::SequenceOfFragments, [949](#), [950](#)
  - gdcm::SequenceOfItems, [958](#)
  - gdcm::StrictScanner, [1027](#)
- EndElement
  - gdcm::TableReader, [1086](#)
  - gdcm::XMLDictReader, [1401](#)
  - gdcm::XMLPrivateDictReader, [1408](#)
- EndElementHandler
  - gdcm::Parser, [769](#)
- EndFilter
  - gdcm::SimpleSubjectWatcher, [985](#)
- EndWith
  - gdcm::Filename, [473](#)
- EnumeratedValues
  - gdcm::EnumeratedValues, [426](#)
- ErrorOff
  - gdcm::Trace, [1114](#)
- ErrorOn
  - gdcm::Trace, [1114](#)
- ErrorType
  - gdcm::Parser, [769](#)
- EstablishConnection
  - gdcm::network::ULConnectionManager, [1216](#)
- EstablishConnectionMove
  - gdcm::network::ULConnectionManager, [1216](#)
- Event
  - gdcm::Event, [428](#)
- Exception
  - gdcm::Exception, [431](#)
- Execute
  - gdcm::Command, [256](#)
  - gdcm::EmptyMaskGenerator, [418](#)
  - gdcm::MemberCommand, [678](#)
  - gdcm::SimpleMemberCommand, [983](#)
- ExecuteData
  - vtkGDCMImageReader, [1276](#)
  - vtkGDCMThreadedImageReader, [1331](#)
- ExecuteInformation
  - vtkGDCMImageReader, [1276](#)
  - vtkGDCMThreadedImageReader, [1331](#)
- ExecuteQuery
  - gdcm::StringFilter, [1038](#)
- Explore
  - gdcm::Directory, [383](#)
- Extract
  - gdcm::IconImageFilter, [510](#)
- ExtractIconImages
  - gdcm::IconImageFilter, [510](#)
- ExtractVeprolconImages
  - gdcm::IconImageFilter, [510](#)
- F

- gdcm::Printer, [856](#)
- gdcm::Reader, [897](#)
- gdcm::Validate, [1243](#)
- gdcm::XMLPrinter, [1406](#)
- Fiducials
  - gdcm::Fiducials, [439](#)
- File
  - gdcm::File, [441](#)
- FileAnonymizer
  - gdcm::FileAnonymizer, [446](#)
- FileChangeTransferSyntax
  - gdcm::FileChangeTransferSyntax, [450](#)
  - gdcm::ImageCodec, [550](#)
- FileDecompressLookupTable
  - gdcm::FileDecompressLookupTable, [453](#)
- FileDerivation
  - gdcm::FileDerivation, [456](#)
- FileExists
  - gdcm::System, [1076](#)
- FileExplicitFilter
  - gdcm::FileExplicitFilter, [460](#)
- FilesDirectory
  - gdcm::System, [1076](#)
- FilesSymlink
  - gdcm::System, [1076](#)
- FileList
  - gdcm, [58](#)
- FileMetaInformation
  - gdcm::FileMetaInformation, [464](#), [465](#)
- FileName
  - vtkGDCMPolyDataReader, [1319](#)
- FileNameEvent
  - gdcm::FileNameEvent, [477](#)
- FileNameOrdering
  - gdcm::SerieHelper, [966](#)
- FileNames
  - vtkGDCMImageReader, [1285](#)
- FileSet
  - gdcm::FileSet, [484](#)
- FileSize
  - gdcm::System, [1076](#)
- FileStreamer
  - gdcm::FileStreamer, [487](#)
- FileTime
  - gdcm::System, [1077](#)
- FileType
  - gdcm::FileSet, [483](#)
- FileWithName
  - gdcm::FileWithName, [491](#)
- Filename
  - gdcm::Filename, [472](#)
- filename
  - gdcm::FileWithName, [492](#)
- FilenameGenerator
  - gdcm::FilenameGenerator, [480](#)
- FilenameType
  - gdcm::DICOMDIRGenerator, [353](#)
  - gdcm::Directory, [383](#)
  - gdcm::FilenameGenerator, [480](#)
- Filenames
  - gdcm::Sorter, [1002](#)
- FilenamesType
  - gdcm::DICOMDIRGenerator, [353](#)
  - gdcm::Directory, [383](#)
  - gdcm::FilenameGenerator, [480](#)
- FileType
  - gdcm::FileSet, [483](#)
- Fill
  - gdcm::ByteValue, [224](#)
- FillFromDataSet
  - gdcm::FileMetaInformation, [466](#)
- FillMedicalImageInformation
  - vtkGDCMImageReader, [1276](#)
  - vtkGDCMImageReader2, [1291](#)
  - vtkGDCMPolyDataReader, [1317](#)
- FindCSAElementByName
  - gdcm::CSAHeader, [286](#)
- FindContext
  - gdcm::network::ULConnection, [1205](#)
- FindDataElement
  - gdcm::DataSet, [329](#), [330](#)
  - gdcm::Item, [603](#)
  - gdcm::SequenceOfItems, [958](#)
- FindDictEntry
  - gdcm::PrivateDict, [858](#)
- FindMacroEntry
  - gdcm::Macro, [658](#)
- FindModuleEntryInMacros
  - gdcm::Module, [695](#)
- FindMrProtocolByName
  - gdcm::MrProtocol, [710](#)
- FindNextDataElement
  - gdcm::DataSet, [330](#)
- FindPDBelementByName
  - gdcm::PDBHeader, [781](#)
- FindPatientRootQuery
  - gdcm::FindPatientRootQuery, [493](#)
- FindStudyRootQuery
  - gdcm::FindStudyRootQuery, [496](#)
- FirstRender
  - vtkImageColorViewer, [1355](#)
- ForceRescale
  - vtkGDCMImageReader, [1285](#)
  - vtkGDCMImageReader2, [1300](#)
- FormatDateTime
  - gdcm::System, [1077](#)
- Fragment
  - gdcm::Fragment, [500](#)



FragmentVector  
     gdcmm::SequenceOfFragments, 948  
 FromString  
     gdcmm::StringFilter, 1039  
  
 GDCM\_DO\_JOIN2  
     gdcmmStaticAssert.h, 1624  
 GDCM\_DO\_JOIN  
     gdcmmStaticAssert.h, 1624  
 GDCM\_EXPORT  
     gdcmmWin32.h, 1680  
 GDCM\_FUNCTION  
     gdcmmTrace.h, 1647  
 GDCM\_JOIN  
     gdcmmStaticAssert.h, 1624  
 GDCM\_LEGACY\_BODY  
     gdcmmLegacyMacro.h, 1531  
 GDCM\_LEGACY\_REPLACED\_BODY  
     gdcmmLegacyMacro.h, 1531  
 GDCM\_LEGACY  
     gdcmmLegacyMacro.h, 1530  
 GDCM\_STATIC\_ASSERT  
     gdcmm::Attribute, 136  
     gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 144  
     gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 153, 154  
     gdcmmStaticAssert.h, 1624  
 GDCMMACROENTRY\_H  
     gdcmmMacroEntry.h, 1536  
 gdcmm, 43  
     AECComp, 58  
     ASComp, 58  
     BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER, 58  
     backslash, 63  
     CSCComp, 58  
     Clamp, 63  
     CompOperators, 60  
     DACComp, 58  
     DTComp, 58  
     ECharSet, 61  
     ENQueryType, 61  
     EQueryLevel, 61  
     EQueryType, 62  
     ERootType, 62  
     FileList, 58  
     GetVRFromTag, 63  
     GlobalInstance, 77  
     IconImage, 59  
     LOComp, 59  
     LTComp, 59  
     LodModeType, 62  
     MacroEntry, 59  
     NestedMacroEntries, 59  
     operator!=, 63  
     operator<<, 63–75  
     operator>>, 75, 76  
     operator==, 75  
     PNComp, 59  
     Round, 76  
     SHComp, 59  
     STComp, 60  
     TMComp, 60  
     TYPETOENCODING, 76  
     to\_string, 76  
     UIComp, 60  
     UTComp, 60  
     VRBINARY, 77  
 gdcmm::ASN1, 130  
     ~ASN1, 131  
     ASN1, 131  
     ParseDump, 131  
     ParseDumpFile, 131  
     TestPBKDF2, 132  
 gdcmm::AbortEvent, 103  
 gdcmm::AnonymizeEvent, 107  
     ~AnonymizeEvent, 108  
     AnonymizeEvent, 108, 109  
     CheckEvent, 109  
     GetEventName, 109  
     GetTag, 109  
     MakeObject, 109  
     Self, 108  
     SetTag, 109  
     Superclass, 108  
 gdcmm::Anonymizer, 110  
     ~Anonymizer, 113  
     Anonymizer, 113  
     BALCPPProtect, 113  
     BasicApplicationLevelConfidentialityProfile, 113  
     CanEmptyTag, 114  
     ClearInternalUIDs, 114  
     Empty, 114  
     GetBasicApplicationLevelConfidentialityProfile←  
         Attributes, 114  
     GetCryptographicMessageSyntax, 115  
     GetFile, 115  
     New, 115  
     RecurseDataSet, 115  
     Remove, 115  
     RemoveGroupLength, 115  
     RemovePrivateTags, 116  
     RemoveRetired, 116  
     Replace, 116  
     SetCryptographicMessageSyntax, 117  
     SetFile, 117  
 gdcmm::AnyEvent, 118  
 gdcmm::ApplicationEntity, 121



- Internal, [123](#)
- IsValid, [122](#)
- MaxLength, [123](#)
- MaxNumberOfComponents, [123](#)
- Padding, [123](#)
- Print, [122](#)
- Separator, [124](#)
- SetBlob, [123](#)
- Squeeze, [123](#)
- gdcmm::Attribute
  - ArrayType, [136](#)
  - GDCM\_STATIC\_ASSERT, [136](#)
  - GetAsDataElement, [137](#)
  - GetDictVM, [137](#)
  - GetDictVR, [137](#)
  - GetNumberOfValues, [137](#)
  - GetTag, [137](#)
  - GetValue, [138](#)
  - GetValues, [138](#)
  - GetVM, [138](#)
  - GetVR, [138](#)
  - Internal, [142](#)
  - operator!=, [139](#)
  - operator<, [139](#)
  - operator==, [139](#)
  - operator[], [139](#)
  - Print, [140](#)
  - Set, [140](#)
  - SetByteValue, [140](#)
  - SetByteValueNoSwap, [140](#)
  - SetFromDataElement, [140](#)
  - SetFromDataSet, [141](#)
  - SetValue, [141](#)
  - SetValues, [141](#)
- gdcmm::Attribute< Group, Element, TVR, TVM >, [134](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [142](#)
  - ArrayType, [143](#)
  - GDCM\_STATIC\_ASSERT, [144](#)
  - GetAsDataElement, [145](#)
  - GetDictVM, [145](#)
  - GetDictVR, [145](#)
  - GetNumberOfValues, [145](#)
  - GetTag, [145](#)
  - GetValue, [145](#), [146](#)
  - GetValues, [146](#)
  - GetVM, [146](#)
  - GetVR, [146](#)
  - Internal, [148](#)
  - operator!=, [146](#)
  - operator<, [146](#)
  - operator==, [147](#)
  - Print, [147](#)
  - Set, [147](#)
  - SetByteValue, [147](#)
  - SetByteValueNoSwap, [147](#)
  - SetFromDataElement, [148](#)
  - SetFromDataSet, [148](#)
  - SetValue, [148](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_3 >, [149](#)
  - GetVM, [150](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_8 >, [150](#)
  - GetVM, [151](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [152](#)
  - ~Attribute, [153](#)
  - ArrayType, [153](#)
  - Attribute, [153](#)
  - GDCM\_STATIC\_ASSERT, [153](#), [154](#)
  - GetAsDataElement, [154](#)
  - GetDictVM, [154](#)
  - GetDictVR, [154](#)
  - GetNumberOfValues, [154](#)
  - GetTag, [155](#)
  - GetValue, [155](#)
  - GetValues, [155](#)
  - GetVM, [155](#)
  - GetVR, [155](#)
  - operator[], [156](#)
  - Print, [156](#)
  - Set, [156](#)
  - SetByteValue, [156](#)
  - SetFromDataElement, [157](#)
  - SetFromDataSet, [157](#)
  - SetNumberOfValues, [157](#)
  - SetValue, [157](#)
  - SetValues, [158](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_2n >, [158](#)
  - GetVM, [159](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2\_n >, [160](#)
  - GetVM, [161](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_3n >, [161](#)
  - GetVM, [162](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM3\_n >, [163](#)
  - GetVM, [164](#)
- gdcmm::AudioCodec, [164](#)
  - ~AudioCodec, [165](#)
  - AudioCodec, [165](#)
  - CanCode, [166](#)
  - CanDecode, [166](#)
  - Decode, [166](#)
- gdcmm::Base64, [167](#)
  - Decode, [167](#)

- Encode, 168
- GetDecodeLength, 168
- GetEncodeLength, 168
- gdcm::BaseQuery, 176
  - ~BaseQuery, 178
  - AddQueryDataSet, 178
  - BaseQuery, 178
  - GetAbstractSyntaxUID, 178
  - GetQueryDataSet, 178, 179
  - GetSOPInstanceUID, 179
  - mDataSet, 181
  - mHelpDescription, 181
  - mSopInstanceUID, 181
  - Print, 179
  - QueryFactory, 181
  - SetSOPInstanceUID, 180
  - SetSearchParameter, 179
  - ValidDataSet, 180
  - ValidateQuery, 180
  - WriteHelpFile, 180
  - WriteQuery, 180
- gdcm::BaseRootQuery, 182
  - ~BaseRootQuery, 183
  - BaseRootQuery, 183
  - Construct, 184
  - GetQueryLevelFromQueryRoot, 184
  - GetQueryLevelFromString, 184
  - GetQueryLevelString, 184
  - GetTagListByLevel, 184
  - InitializeDataSet, 184
  - mHelpDescription, 185
  - mImage, 186
  - mPatient, 186
  - mRootType, 186
  - mSeries, 186
  - mStudy, 186
  - QueryFactory, 185
  - ValidateQuery, 185
- gdcm::BasicOffsetTable, 190
  - BasicOffsetTable, 191
  - operator<<, 191
  - Read, 191
- gdcm::Bitmap, 192
  - ~Bitmap, 195
  - AreOverlaysInPixelFormat, 195
  - Bitmap, 195
  - Clear, 196
  - ComputeLossyFlag, 196
  - Dimensions, 206
  - GetBuffer, 196
  - GetBuffer2, 196
  - GetBufferLength, 196
  - GetColumns, 197
  - GetDataElement, 197
  - GetDimension, 197
  - GetDimensions, 197
  - GetLUT, 198
  - GetNeedByteSwap, 198
  - GetNumberOfDimensions, 198
  - GetPhotometricInterpretation, 198
  - GetPixelFormat, 199
  - GetPlanarConfiguration, 199
  - GetRows, 199
  - GetTransferSyntax, 199
  - ImageChangeTransferSyntax, 205
  - IsEmpty, 200
  - IsLossy, 200
  - IsTransferSyntaxCompatible, 200
  - LUTPtr, 195
  - LUT, 206
  - LossyFlag, 206
  - NeedByteSwap, 206
  - NumberOfDimensions, 206
  - PF, 206
  - PI, 206
  - PixelData, 207
  - PixmapReader, 205
  - PlanarConfiguration, 207
  - Print, 200
  - SetColumns, 200
  - SetDataElement, 201
  - SetDimension, 201
  - SetDimensions, 201
  - SetLUT, 202
  - SetLossyFlag, 201
  - SetNeedByteSwap, 202
  - SetNumberOfDimensions, 202
  - SetPhotometricInterpretation, 202
  - SetPixelFormat, 202
  - SetPlanarConfiguration, 203
  - SetRows, 203
  - SetTransferSyntax, 203
  - TryJPEG2000Codec, 203
  - TryJPEG2000Codec2, 204
  - TryJPEGCodec, 204
  - TryJPEGCodec2, 204
  - TryJPEGLSCodec, 204
  - TryKAKADUCodec, 204
  - TryPVRGCodec, 204
  - TryRAWCodec, 205
  - TryRLECodec, 205
  - TS, 207
  - UnusedBitsPresentInPixelFormat, 205
- gdcm::BitmapToBitmapFilter, 207
  - ~BitmapToBitmapFilter, 209
  - BitmapToBitmapFilter, 209
  - GetOutput, 209
  - GetOutputAsBitmap, 209

- Input, [210](#)
- Output, [210](#)
- SetInput, [209](#)
- gdcmm::BoxRegion, [210](#)
  - ~BoxRegion, [212](#)
  - Area, [212](#)
  - BoundingBox, [212](#)
  - BoxRegion, [212](#)
  - Clone, [213](#)
  - ComputeBoundingBox, [213](#)
  - Empty, [213](#)
  - GetXMax, [213](#)
  - GetXMin, [213](#)
  - GetYMax, [214](#)
  - GetYMin, [214](#)
  - GetZMax, [214](#)
  - GetZMin, [214](#)
  - IsValid, [214](#)
  - operator=, [214](#)
  - Print, [215](#)
  - SetDomain, [215](#)
- gdcmm::ByteBuffer, [215](#)
  - ByteBuffer, [216](#)
  - Get, [216](#)
  - GetStart, [216](#)
  - ShiftEnd, [216](#)
  - UpdatePosition, [216](#)
- gdcmm::ByteSwap
  - Swap, [217](#)
  - SwapFromSwapCodeIntoSystem, [218](#)
  - SwapRange, [218](#)
  - SwapRangeFromSwapCodeIntoSystem, [218](#)
  - SystemIsBigEndian, [218](#)
  - SystemIsLittleEndian, [219](#)
- gdcmm::ByteSwap< T >, [217](#)
- gdcmm::ByteSwapFilter, [219](#)
  - ~ByteSwapFilter, [220](#)
  - ByteSwap, [220](#)
  - ByteSwapFilter, [219](#)
  - SetByteSwapTag, [220](#)
- gdcmm::ByteValue, [221](#)
  - ~ByteValue, [223](#)
  - Append, [223](#)
  - ByteValue, [223](#)
  - Clear, [224](#)
  - ComputeLength, [224](#)
  - Fill, [224](#)
  - GetBuffer, [224](#)
  - GetLength, [224](#)
  - GetPointer, [225](#)
  - IsEmpty, [225](#)
  - IsPrintable, [225](#)
  - operator const std::vector< char > &, [226](#)
  - operator=, [226](#)
  - operator==, [226](#)
  - Print, [226](#)
  - PrintASCIIXML, [227](#)
  - PrintASCII, [226](#)
  - PrintGroupLength, [227](#)
  - PrintHex, [227](#)
  - PrintHexXML, [227](#)
  - PrintPXML, [227](#)
  - Read, [227](#), [228](#)
  - SetLength, [228](#)
  - SetLengthOnly, [228](#)
  - Write, [228](#)
  - WriteBuffer, [229](#)
- gdcmm::CAPICryptoFactory, [229](#)
  - CAPICryptoFactory, [230](#)
  - CreateCMSProvider, [230](#)
- gdcmm::CAPICryptographicMessageSyntax, [231](#)
  - ~CAPICryptographicMessageSyntax, [232](#)
  - CAPICryptographicMessageSyntax, [232](#)
  - Decrypt, [232](#)
  - Encrypt, [232](#)
  - GetCipherType, [233](#)
  - GetInitialized, [233](#)
  - ParseCertificateFile, [233](#)
  - ParseKeyFile, [233](#)
  - SetCipherType, [233](#)
  - SetPassword, [234](#)
- gdcmm::CP246ExplicitDataElement, [267](#)
  - GetLength, [268](#)
  - Read, [268](#)
  - ReadPreValue, [268](#)
  - ReadValue, [268](#)
  - ReadWithLength, [269](#)
- gdcmm::CSAElement, [275](#)
  - CSAElement, [277](#)
  - DataField, [282](#)
  - DataPtr, [277](#)
  - GetByteValue, [278](#)
  - GetKey, [278](#)
  - GetName, [278](#)
  - GetNoOfItems, [278](#)
  - GetSyngoDT, [278](#)
  - GetValue, [279](#)
  - GetVM, [279](#)
  - GetVR, [279](#)
  - IsEmpty, [279](#)
  - KeyField, [282](#)
  - NameField, [282](#)
  - NoOfItemsField, [282](#)
  - operator<, [280](#)
  - operator<<, [282](#)
  - operator=, [280](#)
  - operator==, [280](#)
  - SetByteValue, [280](#)

- SetKey, [280](#)
- SetName, [281](#)
- SetNoOfItems, [281](#)
- SetSyngoDT, [281](#)
- SetValue, [281](#)
- SetVM, [281](#)
- SetVR, [281](#)
- SyngoDTField, [283](#)
- VRField, [283](#)
- ValueMultiplicityField, [283](#)
- gdcmm::CSAHeader, [283](#)
  - ~CSAHeader, [286](#)
  - CSAHeader, [286](#)
  - CSAHeaderType, [285](#)
  - FindCSAElementByName, [286](#)
  - GetCSADataInfo, [286](#)
  - GetCSAEEnd, [286](#)
  - GetCSAElementByName, [287](#)
  - GetCSAImageHeaderInfoTag, [287](#)
  - GetCSASeriesHeaderInfoTag, [287](#)
  - GetDataSet, [287](#)
  - GetFormat, [288](#)
  - GetInterfile, [288](#)
  - GetMrProtocol, [288](#)
  - LoadFromDataElement, [288](#)
  - operator<<, [289](#)
  - Print, [288](#)
  - Read, [289](#)
  - Write, [289](#)
- gdcmm::CSAHeaderDict, [290](#)
  - AddCSAHeaderDictEntry, [291](#)
  - Begin, [291](#)
  - CSAHeaderDict, [291](#)
  - ConstIterator, [290](#)
  - Dicts, [292](#)
  - End, [291](#)
  - GetCSAHeaderDictEntry, [292](#)
  - IsEmpty, [292](#)
  - Iterator, [291](#)
  - LoadDefault, [292](#)
  - MapCSAHeaderDictEntry, [291](#)
  - operator<<, [292](#)
- gdcmm::CSAHeaderDictEntry, [293](#)
  - CSAHeaderDictEntry, [294](#)
  - GetDescription, [294](#)
  - GetName, [294](#)
  - GetVM, [294](#)
  - GetVR, [294](#)
  - operator<, [295](#)
  - operator<<, [296](#)
  - SetDescription, [295](#)
  - SetName, [295](#)
  - SetVM, [295](#)
  - SetVR, [295](#)
- gdcmm::CSAHeaderDictException, [296](#)
- gdcmm::CodeString, [249](#)
  - CodeString, [252](#)
  - const\_iterator, [250](#)
  - const\_reference, [251](#)
  - const\_reverse\_iterator, [251](#)
  - difference\_type, [251](#)
  - GetAsString, [253](#)
  - IsValid, [253](#)
  - iterator, [251](#)
  - operator!=, [253](#)
  - operator<<, [254](#)
  - operator==, [254](#)
  - pointer, [251](#)
  - reference, [251](#)
  - reverse\_iterator, [251](#)
  - Size, [253](#)
  - size\_type, [252](#)
  - TrimInternal, [253](#)
  - value\_type, [252](#)
- gdcmm::Codec, [246](#)
- gdcmm::Coder, [247](#)
  - ~Coder, [248](#)
  - CanCode, [248](#)
  - Code, [248](#)
  - InternalCode, [249](#)
- gdcmm::Command, [254](#)
  - ~Command, [256](#)
  - Command, [255](#)
  - Execute, [256](#)
- gdcmm::CommandDataSet, [257](#)
  - ~CommandDataSet, [258](#)
  - CommandDataSet, [258](#)
  - Insert, [258](#)
  - operator<<, [259](#)
  - Read, [258](#)
  - Replace, [259](#)
  - Write, [259](#)
- gdcmm::CompositeNetworkFunctions, [261](#)
  - CEcho, [263](#)
  - CFind, [263](#)
  - CMove, [264](#)
  - CStore, [265](#)
  - ConstructQuery, [264](#), [265](#)
  - KeyValuePairArrayType, [262](#)
  - KeyValuePairType, [262](#)
- gdcmm::ConstCharWrapper, [266](#)
  - ConstCharWrapper, [266](#)
  - operator const char \*, [266](#)
- gdcmm::CryptoFactory, [269](#)
  - ~CryptoFactory, [271](#)
  - CreateCMSProvider, [271](#)
  - CryptoFactory, [271](#)
  - CryptoLib, [270](#)

- GetFactoryInstance, 271
- gdcm::CryptographicMessageSyntax, 272
  - ~CryptographicMessageSyntax, 273
  - CipherTypes, 272
  - CryptographicMessageSyntax, 273
  - Decrypt, 273
  - Encrypt, 273
  - GetCipherType, 274
  - ParseCertificateFile, 274
  - ParseKeyFile, 274
  - SetCipherType, 274
  - SetPassword, 275
- gdcm::Curve, 300
  - ~Curve, 302
  - Curve, 301, 302
  - Decode, 302
  - GetAsPoints, 302
  - GetCurveDataDescriptor, 302
  - GetDataValueRepresentation, 302
  - GetDimensions, 303
  - GetGroup, 303
  - GetNumberOfCurves, 303
  - GetNumberOfPoints, 303
  - GetTypeOfData, 303
  - GetTypeOfDataDescription, 303
  - IsEmpty, 303
  - Print, 304
  - SetCoordinateStartValue, 304
  - SetCoordinateStepValue, 304
  - SetCurve, 304
  - SetCurveDataDescriptor, 304
  - SetCurveDescription, 304
  - SetDataValueRepresentation, 305
  - SetDimensions, 305
  - SetGroup, 305
  - SetNumberOfPoints, 305
  - SetTypeOfData, 305
  - Update, 305
- gdcm::DICOMDIRGenerator, 352
  - ~DICOMDIRGenerator, 354
  - AddImageDirectoryRecord, 354
  - AddPatientDirectoryRecord, 354
  - AddSeriesDirectoryRecord, 354
  - AddStudyDirectoryRecord, 354
  - DICOMDIRGenerator, 354
  - FilenameType, 353
  - FilenamesType, 353
  - Generate, 355
  - GetFile, 355
  - GetScanner, 355
  - SetDescriptor, 355
  - SetFile, 355
  - SetFilenames, 355
  - SetRootDirectory, 356
- gdcm::DICOMDIR, 351
  - DICOMDIR, 351, 352
- gdcm::DataElement, 306
  - Clear, 310
  - DataElement, 309, 310
  - Empty, 310
  - GetByteValue, 310
  - GetLength, 311
  - GetSequenceOfFragments, 311
  - GetTag, 312
  - GetValue, 312
  - GetValueAsSQ, 313
  - GetVL, 313
  - GetVR, 313
  - IsEmpty, 314
  - IsUndefinedLength, 314
  - operator<, 314
  - operator<<, 319
  - operator=, 315
  - operator==, 315
  - Read, 315
  - ReadOrSkip, 315
  - ReadPreValue, 315
  - ReadValue, 316
  - ReadValueWithLength, 316
  - ReadWithLength, 316
  - SetByteValue, 316
  - SetTag, 317
  - SetVLToUndefined, 318
  - SetValue, 317
  - SetValueFieldLength, 318
  - SetVL, 318
  - SetVR, 318
  - TagField, 319
  - VRField, 320
  - ValueField, 320
  - ValueLengthField, 320
  - ValuePtr, 309
  - Write, 319
- gdcm::DataElementException, 321
- gdcm::DataEvent, 321
  - ~DataEvent, 323
  - CheckEvent, 324
  - DataEvent, 323, 324
  - GetData, 324
  - GetDataLength, 324
  - GetEventName, 324
  - MakeObject, 324
  - Self, 323
  - SetData, 325
  - Superclass, 323
- gdcm::DataSet, 325
  - Begin, 328
  - CSAHeader, 337

- Clear, [329](#)
- ComputeDataElement, [329](#)
- ComputeGroupLength, [329](#)
- ConstIterator, [328](#)
- DataElementSet, [328](#)
- End, [329](#)
- FindDataElement, [329](#), [330](#)
- FindNextDataElement, [330](#)
- GetDEEnd, [331](#)
- GetDES, [331](#)
- GetDataElement, [330](#), [331](#)
- GetLength, [332](#)
- GetMediaStorage, [332](#)
- GetPrivateCreator, [332](#)
- Insert, [332](#)
- InsertDataElement, [332](#)
- IsEmpty, [333](#)
- Iterator, [328](#)
- operator<<, [337](#)
- operator(), [333](#)
- operator=, [333](#)
- operator[], [333](#)
- Print, [333](#)
- Read, [334](#)
- ReadNested, [334](#)
- ReadSelectedPrivateTags, [334](#)
- ReadSelectedPrivateTagsWithLength, [334](#)
- ReadSelectedTags, [334](#)
- ReadSelectedTagsWithLength, [335](#)
- ReadUpToTag, [335](#)
- ReadUpToTagWithLength, [335](#)
- ReadWithLength, [335](#)
- Remove, [335](#)
- Replace, [336](#)
- ReplaceEmpty, [336](#)
- Size, [336](#)
- SizeType, [328](#)
- Write, [337](#)
- gdcmm::DataSetEvent, [338](#)
  - ~DataSetEvent, [339](#)
  - CheckEvent, [340](#)
  - DataSetEvent, [339](#), [340](#)
  - GetDataSet, [340](#)
  - GetEventName, [340](#)
  - MakeObject, [340](#)
  - Self, [339](#)
  - Superclass, [339](#)
- gdcmm::DataSetHelper, [341](#)
  - ComputeVR, [341](#)
- gdcmm::Decoder, [342](#)
  - ~Decoder, [342](#)
  - CanDecode, [343](#)
  - Decode, [343](#)
  - DecodeByStreams, [343](#)
- gdcmm::DefinedTerms, [344](#)
  - DefinedTerms, [344](#)
- gdcmm::Defs, [344](#)
  - ~Defs, [346](#)
  - Defs, [345](#)
  - GetIODFromFile, [346](#)
  - GetIODNameFromMediaStorage, [346](#)
  - GetIODs, [346](#)
  - GetMacros, [347](#)
  - GetModules, [347](#)
  - GetTypeFromTag, [347](#)
  - Global, [348](#)
  - IsEmpty, [348](#)
  - LoadDefaults, [348](#)
  - LoadFromFile, [348](#)
  - Verify, [348](#)
- gdcmm::DeltaEncodingCodec, [349](#)
  - ~DeltaEncodingCodec, [350](#)
  - CanDecode, [350](#)
  - Decode, [350](#), [351](#)
  - DeltaEncodingCodec, [350](#)
- gdcmm::Dict, [356](#)
  - AddDictEntry, [358](#)
  - Begin, [358](#)
  - ConstIterator, [357](#)
  - Dict, [358](#)
  - Dicts, [360](#)
  - End, [358](#)
  - GetDictEntry, [358](#)
  - GetDictEntryByKeyword, [359](#)
  - GetDictEntryByName, [359](#)
  - GetKeywordFromTag, [359](#)
  - IsEmpty, [359](#)
  - Iterator, [357](#)
  - LoadDefault, [360](#)
  - MapDictEntry, [357](#)
  - operator<<, [360](#)
- gdcmm::DictConverter, [360](#)
  - ~DictConverter, [362](#)
  - AddGroupLength, [362](#)
  - Convert, [362](#)
  - ConvertToCXX, [362](#)
  - ConvertToXML, [362](#)
  - DictConverter, [362](#)
  - GetDictName, [363](#)
  - GetInputFilename, [363](#)
  - GetOutputFilename, [363](#)
  - GetOutputType, [363](#)
  - OutputTypes, [361](#)
  - ReadVM, [363](#)
  - ReadVR, [364](#)
  - Readuint16, [363](#)
  - SetDictName, [364](#)
  - SetInputFileName, [364](#)

- SetOutputFileName, [364](#)
- SetOutputType, [364](#)
- WriteFooter, [364](#)
- WriteHeader, [365](#)
- gdcm::DictEntry, [365](#)
  - Dict, [369](#)
  - DictEntry, [366](#)
  - GetKeyword, [366](#)
  - GetName, [367](#)
  - GetRetired, [367](#)
  - GetVM, [367](#)
  - GetVR, [367](#)
  - IsUnique, [368](#)
  - operator<<, [369](#)
  - SetElementXX, [368](#)
  - SetGroupXX, [368](#)
  - SetKeyword, [368](#)
  - SetName, [368](#)
  - SetRetired, [369](#)
  - SetVM, [369](#)
  - SetVR, [369](#)
- gdcm::DictPrinter, [370](#)
  - ~DictPrinter, [371](#)
  - DictPrinter, [371](#)
  - Print, [371](#)
  - PrintDataElement2, [371](#)
  - PrintDataSet2, [372](#)
- gdcm::Dicts, [372](#)
  - ~Dicts, [374](#)
  - ConstructorType, [373](#)
  - Dicts, [373](#)
  - GetCSAHeaderDict, [374](#)
  - GetConstructorString, [374](#)
  - GetDictEntry, [374](#)
  - GetPrivateDict, [375](#)
  - GetPublicDict, [375](#)
  - Global, [376](#)
  - IsEmpty, [375](#)
  - LoadDefaults, [375](#)
  - operator<<, [376](#)
- gdcm::DirectionCosines, [378](#)
  - ~DirectionCosines, [379](#)
  - ComputeDistAlongNormal, [379](#)
  - Cross, [379](#)
  - CrossDot, [379](#)
  - DirectionCosines, [379](#)
  - Dot, [380](#)
  - IsValid, [380](#)
  - Normalize, [380](#)
  - operator const double \*, [381](#)
  - Print, [381](#)
  - SetFromString, [381](#)
- gdcm::Directory, [381](#)
  - ~Directory, [383](#)
  - Directory, [383](#)
  - Explore, [383](#)
  - FilenameType, [383](#)
  - FileNamesType, [383](#)
  - GetDirectories, [384](#)
  - GetFilenames, [384](#)
  - GetToplevel, [384](#)
  - Load, [384](#)
  - operator<<, [385](#)
  - Print, [385](#)
- gdcm::DirectoryHelper, [386](#)
  - GetCTImageSeriesUIDs, [386](#)
  - GetFilenamesFromSeriesUIDs, [386](#)
  - GetFrameOfReference, [387](#)
  - GetMRIImageSeriesUIDs, [387](#)
  - GetRTStructSeriesUIDs, [387](#)
  - GetSOPClassUID, [387](#)
  - GetSeriesUIDsBySOPClassUID, [387](#)
  - GetStringValueFromTag, [388](#)
  - LoadImageFromFiles, [388](#)
  - RetrieveSOPInstanceUIDFromIndex, [388](#)
  - RetrieveSOPInstanceUIDFromZPosition, [388](#)
- gdcm::DummyValueGenerator, [388](#)
  - Generate, [389](#)
- gdcm::Dumper, [389](#)
  - ~Dumper, [391](#)
  - Dumper, [390](#)
- gdcm::Element
  - GetAsDataElement, [393](#)
  - GetLength, [393](#)
  - GetValue, [394](#)
  - GetValues, [394](#)
  - GetVM, [394](#)
  - GetVR, [394](#)
  - Internal, [396](#)
  - operator[], [394](#)
  - Print, [395](#)
  - Read, [395](#)
  - Set, [395](#)
  - SetFromDataElement, [395](#)
  - SetNoSwap, [395](#)
  - SetValue, [395](#)
  - Type, [393](#)
  - Write, [396](#)
- gdcm::Element< TVR, TVM >, [391](#)
- gdcm::Element< TVR, VM::VM1\_2 >, [396](#)
  - Parent, [397](#)
  - SetLength, [397](#)
- gdcm::Element< TVR, VM::VM1\_n >, [398](#)
  - ~Element, [399](#)
  - Element, [399](#)
  - GetAsDataElement, [400](#)
  - GetLength, [400](#)
  - GetValue, [400](#)

- GetVM, [400](#)
- GetVR, [400](#)
- operator=, [401](#)
- operator[], [401](#)
- Print, [401](#)
- Read, [401](#)
- Set, [401](#)
- SetArray, [402](#)
- SetFromDataElement, [402](#)
- SetLength, [402](#)
- SetNoSwap, [402](#)
- SetValue, [402](#)
- Type, [399](#)
- Write, [403](#)
- WriteASCII, [403](#)
- gdcmm::Element< TVR, VM::VM2\_2n >, [403](#)
  - Parent, [404](#)
  - SetLength, [404](#)
- gdcmm::Element< TVR, VM::VM2\_n >, [405](#)
  - Parent, [406](#)
  - SetLength, [406](#)
- gdcmm::Element< TVR, VM::VM3\_3n >, [407](#)
  - Parent, [408](#)
  - SetLength, [408](#)
- gdcmm::Element< TVR, VM::VM3\_n >, [409](#)
  - Parent, [410](#)
  - SetLength, [410](#)
- gdcmm::Element< VR::AS, VM::VM5 >, [411](#)
  - GetLength, [411](#)
  - Internal, [411](#)
  - Print, [411](#)
- gdcmm::Element< VR::OB, VM::VM1 >, [412](#)
- gdcmm::Element< VR::OW, VM::VM1 >, [413](#)
- gdcmm::ElementDisableCombinations< TVR, TVM >, [415](#)
- gdcmm::ElementDisableCombinations< VR::OB, VM::V←  
M1\_n >, [416](#)
- gdcmm::ElementDisableCombinations< VR::OW, VM::V←  
M1\_n >, [416](#)
- gdcmm::EmptyMaskGenerator, [416](#)
  - ~EmptyMaskGenerator, [418](#)
  - EmptyMaskGenerator, [418](#)
  - Execute, [418](#)
  - SOPClassUIDMode, [418](#)
  - SetInputDirectory, [418](#)
  - SetOutputDirectory, [419](#)
  - SetSOPClassUIDMode, [419](#)
- gdcmm::EncapsulatedDocument, [419](#)
  - EncapsulatedDocument, [420](#)
- gdcmm::EncodingImplementation< T >, [420](#)
- gdcmm::EncodingImplementation< VR::VRASCII >, [421](#)
  - Read, [421](#)
  - ReadComputeLength, [421](#)
  - ReadNoSwap, [421](#)
  - Write, [422](#)
- gdcmm::EncodingImplementation< VR::VRBINARY >, [423](#)
  - Read, [423](#)
  - ReadComputeLength, [423](#)
  - ReadNoSwap, [423](#)
  - Write, [423](#)
- gdcmm::EndEvent, [424](#)
- gdcmm::EnumeratedValues, [425](#)
  - EnumeratedValues, [426](#)
- gdcmm::Event, [426](#)
  - ~Event, [428](#)
  - CheckEvent, [428](#)
  - Event, [428](#)
  - GetEventName, [428](#)
  - MakeObject, [429](#)
  - Print, [429](#)
- gdcmm::Exception, [430](#)
  - ~Exception, [431](#)
  - Exception, [431](#)
  - GetDescription, [431](#)
  - what, [431](#)
- gdcmm::ExitEvent, [432](#)
- gdcmm::ExplicitDataElement, [433](#)
  - GetLength, [435](#)
  - Read, [435](#)
  - ReadPreValue, [435](#)
  - ReadValue, [435](#)
  - ReadWithLength, [435](#)
  - Write, [435](#)
- gdcmm::ExplicitImplicitDataElement, [436](#)
  - GetLength, [438](#)
  - Read, [438](#)
  - ReadPreValue, [438](#)
  - ReadValue, [438](#)
  - ReadWithLength, [438](#)
- gdcmm::Fiducials, [439](#)
  - Fiducials, [439](#)
- gdcmm::File, [439](#)
  - ~File, [441](#)
  - File, [441](#)
  - GetDataSet, [442](#)
  - GetHeader, [442](#)
  - operator<<, [443](#)
  - Read, [443](#)
  - SetDataSet, [443](#)
  - SetHeader, [443](#)
  - Write, [443](#)
- gdcmm::FileAnonymizer, [444](#)
  - ~FileAnonymizer, [446](#)
  - Empty, [446](#)
  - FileAnonymizer, [446](#)
  - Remove, [446](#)
  - Replace, [447](#)
  - SetInputFileName, [447](#)
  - SetOutputFileName, [447](#)



- Write, [448](#)
- gdcmm::FileChangeTransferSyntax, [448](#)
  - ~FileChangeTransferSyntax, [450](#)
  - Change, [450](#)
  - FileChangeTransferSyntax, [450](#)
  - GetCodec, [450](#)
  - New, [450](#)
  - SetInputFileName, [451](#)
  - SetOutputFileName, [451](#)
  - SetTransferSyntax, [451](#)
- gdcmm::FileDecompressLookupTable, [452](#)
  - ~FileDecompressLookupTable, [453](#)
  - Change, [453](#)
  - FileDecompressLookupTable, [453](#)
  - GetFile, [454](#)
  - GetPixmap, [454](#)
  - SetFile, [454](#)
  - SetPixmap, [454](#)
- gdcmm::FileDerivation, [455](#)
  - ~FileDerivation, [456](#)
  - AddDerivationDescription, [456](#)
  - AddPurposeOfReferenceCodeSequence, [456](#)
  - AddReference, [456](#)
  - AddSourceImageSequence, [456](#)
  - Derive, [457](#)
  - FileDerivation, [456](#)
  - GetFile, [457](#)
  - SetAppendDerivationHistory, [457](#)
  - SetDerivationCodeSequenceCodeValue, [457](#)
  - SetDerivationDescription, [458](#)
  - SetFile, [458](#)
  - SetPurposeOfReferenceCodeSequenceCodeValue, [458](#)
- gdcmm::FileExplicitFilter, [459](#)
  - ~FileExplicitFilter, [460](#)
  - Change, [460](#)
  - ChangeFMI, [460](#)
  - FileExplicitFilter, [460](#)
  - GetFile, [460](#)
  - ProcessDataSet, [460](#)
  - SetChangePrivateTags, [461](#)
  - SetFile, [461](#)
  - SetRecomputeItemLength, [461](#)
  - SetRecomputeSequenceLength, [461](#)
  - SetUseVRUN, [461](#)
- gdcmm::FileMetaInformation, [462](#)
  - ~FileMetaInformation, [465](#)
  - AppendImplementationClassUID, [465](#)
  - ComputeDataSetMediaStorageSOPClass, [465](#)
  - ComputeDataSetTransferSyntax, [465](#)
  - DataSetMS, [471](#)
  - DataSetTS, [471](#)
  - Default, [465](#)
  - FileMetaInformation, [464](#), [465](#)
  - FillFromDataSet, [466](#)
  - GetDataSetTransferSyntax, [466](#)
  - GetFileMetaInformationVersion, [466](#)
  - GetFullLength, [466](#)
  - GetGDCMImplementationClassUID, [466](#)
  - GetGDCMImplementationVersionName, [466](#)
  - GetGDCMSourceApplicationEntityTitle, [467](#)
  - GetImplementationClassUID, [467](#)
  - GetImplementationVersionName, [467](#)
  - GetMediaStorage, [467](#)
  - GetMediaStorageAsString, [467](#)
  - GetMetaInformationTS, [467](#)
  - GetPreamble, [467](#), [468](#)
  - GetSourceApplicationEntityTitle, [468](#)
  - Insert, [468](#)
  - IsValid, [468](#)
  - MetaInformationTS, [471](#)
  - operator<<, [470](#)
  - Read, [468](#)
  - ReadCompat, [468](#)
  - ReadCompatInternal, [469](#)
  - Replace, [469](#)
  - SetDataSetTransferSyntax, [469](#)
  - SetImplementationClassUID, [469](#)
  - SetImplementationVersionName, [470](#)
  - SetPreamble, [470](#)
  - SetSourceApplicationEntityTitle, [470](#)
  - Write, [470](#)
- gdcmm::FileNameEvent, [475](#)
  - ~FileNameEvent, [477](#)
  - CheckEvent, [477](#)
  - FileNameEvent, [477](#)
  - GetEventName, [478](#)
  - GetFileName, [478](#)
  - MakeObject, [478](#)
  - Self, [477](#)
  - SetFileName, [478](#)
  - Superclass, [477](#)
- gdcmm::FileSet, [483](#)
  - AddFile, [484](#)
  - FileSet, [484](#)
  - FileType, [483](#)
  - FilesType, [483](#)
  - GetFiles, [484](#)
  - operator<<, [485](#)
  - SetFiles, [484](#)
- gdcmm::FileStreamer, [485](#)
  - ~FileStreamer, [487](#)
  - AppendToDataElement, [487](#)
  - AppendToGroupDataElement, [487](#)
  - CheckDataElement, [488](#)
  - CheckTemplateFileName, [488](#)
  - FileStreamer, [487](#)
  - New, [488](#)

- ReserveDataElement, [488](#)
- ReserveGroupDataElement, [488](#)
- SetOutputFileName, [489](#)
- SetTemplateFileName, [489](#)
- StartDataElement, [489](#)
- StartGroupDataElement, [489](#)
- StopDataElement, [490](#)
- StopGroupDataElement, [490](#)
- gdcmm::FileWithName, [490](#)
  - FileWithName, [491](#)
  - filename, [492](#)
- gdcmm::Filename, [471](#)
  - EndWith, [473](#)
  - Filename, [472](#)
  - GetExtension, [473](#)
  - GetFileName, [473](#)
  - GetName, [473](#)
  - GetPath, [473](#)
  - IsEmpty, [473](#)
  - IsIdentical, [474](#)
  - Join, [474](#)
  - operator const char \*, [474](#)
  - ToUnixSlashes, [474](#)
  - ToWindowsSlashes, [474](#)
- gdcmm::FilenameGenerator, [479](#)
  - ~FilenameGenerator, [480](#)
  - FilenameGenerator, [480](#)
  - FilenameType, [480](#)
  - FilenamesType, [480](#)
  - Generate, [480](#)
  - GetFilename, [481](#)
  - GetFilenames, [481](#)
  - GetNumberOfFilenames, [481](#)
  - GetPattern, [481](#)
  - GetPrefix, [482](#)
  - SetNumberOfFilenames, [482](#)
  - SetPattern, [482](#)
  - SetPrefix, [482](#)
  - SizeType, [480](#)
- gdcmm::FindPatientRootQuery, [492](#)
  - FindPatientRootQuery, [493](#)
  - GetAbstractSyntaxUID, [494](#)
  - GetTagListByLevel, [494](#)
  - InitializeDataSet, [494](#)
  - QueryFactory, [495](#)
  - ValidateQuery, [494](#)
- gdcmm::FindStudyRootQuery, [495](#)
  - FindStudyRootQuery, [496](#)
  - GetAbstractSyntaxUID, [497](#)
  - GetTagListByLevel, [497](#)
  - InitializeDataSet, [497](#)
  - QueryFactory, [498](#)
  - ValidateQuery, [497](#)
- gdcmm::Fragment, [498](#)
  - ComputeLength, [500](#)
  - Fragment, [500](#)
  - GetLength, [500](#)
  - operator<<, [501](#)
  - Read, [500](#)
  - ReadBacktrack, [500](#)
  - ReadPreValue, [500](#)
  - ReadValue, [501](#)
  - Write, [501](#)
- gdcmm::Global, [501](#)
  - ~Global, [503](#)
  - Append, [503](#)
  - GetDefs, [503](#)
  - GetDicts, [503](#), [504](#)
  - GetInstance, [504](#)
  - Global, [502](#)
  - LoadResourcesFiles, [504](#)
  - Locate, [504](#)
  - operator<<, [505](#)
  - Prepend, [505](#)
- gdcmm::GroupDict, [505](#)
  - ~GroupDict, [506](#)
  - Add, [507](#)
  - GetAbbreviation, [507](#)
  - GetName, [507](#)
  - GroupDict, [506](#)
  - GroupStringVector, [506](#)
  - Insert, [507](#)
  - operator<<, [508](#)
  - Size, [507](#)
- gdcmm::IODEntry, [590](#)
  - GetIE, [591](#)
  - GetName, [591](#)
  - GetRef, [591](#)
  - GetUsage, [591](#)
  - GetUsageType, [591](#)
  - IODEntry, [591](#)
  - operator<<, [592](#)
  - SetIE, [592](#)
  - SetName, [592](#)
  - SetRef, [592](#)
  - SetUsage, [592](#)
- gdcmm::IODs, [593](#)
  - AddIOD, [594](#)
  - Begin, [594](#)
  - Clear, [595](#)
  - End, [595](#)
  - GetIOD, [595](#)
  - IODMapType, [593](#)
  - IODMapTypeConstIterator, [594](#)
  - IODName, [594](#)
  - IODs, [594](#)
  - operator<<, [595](#)
- gdcmm::IOD, [587](#)

- AddIODEntry, [588](#)
- Clear, [588](#)
- GetIODEntry, [588](#)
- GetNumberOfIODs, [589](#)
- GetTypeFromTag, [589](#)
- IOD, [588](#)
- MapIODEntry, [587](#)
- operator<<, [589](#)
- SizeType, [588](#)
- gdcm::IPPSorter, [596](#)
  - ComputeZSpacing, [600](#)
  - DirCosTolerance, [600](#)
  - DropDuplicatePositions, [600](#)
  - GetDirectionCosinesTolerance, [598](#)
  - GetZSpacing, [598](#)
  - GetZSpacingTolerance, [598](#)
  - IPPSorter, [597](#)
  - SetComputeZSpacing, [598](#)
  - SetDirectionCosinesTolerance, [598](#)
  - SetDropDuplicatePositions, [599](#)
  - SetZSpacingTolerance, [599](#)
  - Sort, [599](#)
  - ZSpacing, [600](#)
  - ZTolerance, [600](#)
- gdcm::IconImageFilter, [508](#)
  - ~IconImageFilter, [509](#)
  - Extract, [510](#)
  - ExtractIconImages, [510](#)
  - ExtractVeprolIconImages, [510](#)
  - GetFile, [510](#)
  - GetIconImage, [510](#)
  - GetNumberOfIconImages, [511](#)
  - IconImageFilter, [509](#)
  - SetFile, [511](#)
- gdcm::IconImageGenerator, [511](#)
  - ~IconImageGenerator, [513](#)
  - AutoPixelMinMax, [513](#)
  - ConvertRGBToPaletteColor, [513](#)
  - Generate, [513](#)
  - GetIconImage, [513](#)
  - GetPixmap, [514](#)
  - IconImageGenerator, [512](#)
  - SetOutputDimensions, [514](#)
  - SetOutsideValuePixel, [514](#)
  - SetPixelMinMax, [514](#)
  - SetPixmap, [515](#)
- gdcm::Image, [516](#)
  - ~Image, [518](#)
  - GetDirectionCosines, [519](#)
  - GetIntercept, [519](#)
  - GetOrigin, [519](#)
  - GetSlope, [520](#)
  - GetSpacing, [520](#)
  - Image, [518](#)
  - Print, [520](#)
  - SetDirectionCosines, [520](#), [521](#)
  - SetIntercept, [521](#)
  - SetOrigin, [521](#), [522](#)
  - SetSlope, [522](#)
  - SetSpacing, [522](#)
- gdcm::ImageApplyLookupTable, [523](#)
  - ~ImageApplyLookupTable, [525](#)
  - Apply, [525](#)
  - ImageApplyLookupTable, [525](#)
- gdcm::ImageChangePhotometricInterpretation, [526](#)
  - ~ImageChangePhotometricInterpretation, [528](#)
  - Change, [528](#)
  - ChangeMonochrome, [529](#)
  - ChangeRGB2YBR, [529](#)
  - ChangeYBR2RGB, [529](#)
  - GetPhotometricInterpretation, [529](#)
  - ImageChangePhotometricInterpretation, [528](#)
  - RGB2YBR, [529](#)
  - SetPhotometricInterpretation, [529](#)
  - YBR2RGB, [530](#)
- gdcm::ImageChangePlanarConfiguration, [530](#)
  - ~ImageChangePlanarConfiguration, [532](#)
  - Change, [532](#)
  - GetPlanarConfiguration, [532](#)
  - ImageChangePlanarConfiguration, [532](#)
  - RGBPixelsToRGBPlanes, [533](#)
  - RGBPlanesToRGBPixels, [533](#)
  - SetPlanarConfiguration, [533](#)
- gdcm::ImageChangeTransferSyntax, [534](#)
  - ~ImageChangeTransferSyntax, [536](#)
  - Change, [537](#)
  - GetTransferSyntax, [537](#)
  - ImageChangeTransferSyntax, [536](#)
  - SetCompressIconImage, [537](#)
  - SetForce, [537](#)
  - SetTransferSyntax, [537](#)
  - SetUserCodec, [538](#)
  - TryJPEG2000Codec, [538](#)
  - TryJPEGCodec, [538](#)
  - TryJPEGLSCodec, [538](#)
  - TryRAWCodec, [539](#)
  - TryRLECodec, [539](#)
- gdcm::ImageCodec, [539](#)
  - ~ImageCodec, [542](#)
  - AppendFrameEncode, [542](#)
  - AppendRowEncode, [542](#)
  - CanCode, [543](#)
  - CanDecode, [543](#)
  - CleanupUnusedBits, [543](#)
  - Clone, [543](#)
  - Decode, [544](#)
  - DecodeByStreams, [544](#)
  - Dimensions, [551](#)

- DoByteSwap, [544](#)
- DoInvertMonochrome, [544](#)
- DoOverlayCleanup, [545](#)
- DoPaddedCompositePixelCode, [545](#)
- DoPlanarConfiguration, [545](#)
- DoSimpleCopy, [545](#)
- DoYBR, [545](#)
- FileChangeTransferSyntax, [550](#)
- GetDimensions, [545](#)
- GetHeaderInfo, [546](#)
- GetLUT, [546](#)
- GetLossyFlag, [546](#)
- GetNeedByteSwap, [546](#)
- GetNumberOfDimensions, [546](#)
- GetPhotometricInterpretation, [546](#)
- GetPixelFormat, [547](#)
- GetPlanarConfiguration, [547](#)
- ImageChangePhotometricInterpretation, [551](#)
- ImageCodec, [542](#)
- IsFrameEncoder, [547](#)
- IsLossy, [547](#)
- IsRowEncoder, [547](#)
- IsValid, [548](#)
- LUTPtr, [542](#)
- LUT, [551](#)
- LossyFlag, [551](#)
- NeedByteSwap, [551](#)
- NeedOverlayCleanup, [551](#)
- NumberOfDimensions, [552](#)
- PF, [552](#)
- PI, [552](#)
- PlanarConfiguration, [552](#)
- RequestPaddedCompositePixelCode, [552](#)
- RequestPlanarConfiguration, [552](#)
- SetDimensions, [548](#)
- SetLUT, [548](#)
- SetLossyFlag, [548](#)
- SetNeedByteSwap, [549](#)
- SetNeedOverlayCleanup, [549](#)
- SetNumberOfDimensions, [549](#)
- SetPhotometricInterpretation, [549](#)
- SetPixelFormat, [549](#)
- SetPlanarConfiguration, [550](#)
- StartEncode, [550](#)
- StopEncode, [550](#)
- gdcm::ImageConverter, [553](#)
  - ~ImageConverter, [553](#)
  - Convert, [553](#)
  - GetOutput, [554](#)
  - ImageConverter, [553](#)
  - SetInput, [554](#)
- gdcm::ImageFragmentSplitter, [554](#)
  - ~ImageFragmentSplitter, [556](#)
  - GetFragmentSizeMax, [556](#)
  - ImageFragmentSplitter, [556](#)
  - SetForce, [556](#)
  - SetFragmentSizeMax, [556](#)
  - Split, [557](#)
- gdcm::ImageHelper, [557](#)
  - ComputeMediaStorageFromModality, [558](#)
  - ComputeSpacingFromImagePositionPatient, [558](#)
  - GetDimensionsValue, [559](#)
  - GetDirectionCosinesFromDataSet, [559](#)
  - GetDirectionCosinesValue, [559](#)
  - GetForcePixelSpacing, [559](#)
  - GetForceRescaleInterceptSlope, [559](#)
  - GetLUT, [560](#)
  - GetOriginValue, [560](#)
  - GetPMSRescaleInterceptSlope, [560](#)
  - GetPhotometricInterpretationValue, [560](#)
  - GetPixelFormatValue, [560](#)
  - GetPlanarConfigurationValue, [560](#)
  - GetPointerFromElement, [561](#)
  - GetRealWorldValueMappingContent, [561](#)
  - GetRescaleInterceptSlopeValue, [561](#)
  - GetSpacingTagFromMediaStorage, [561](#)
  - GetSpacingValue, [561](#)
  - GetZSpacingTagFromMediaStorage, [562](#)
  - SetDimensionsValue, [562](#)
  - SetDirectionCosinesValue, [562](#)
  - SetForcePixelSpacing, [562](#)
  - SetForceRescaleInterceptSlope, [562](#)
  - SetOriginValue, [563](#)
  - SetPMSRescaleInterceptSlope, [563](#)
  - SetRescaleInterceptSlopeValue, [563](#)
  - SetSpacingValue, [563](#)
- gdcm::ImageReader, [564](#)
  - ~ImageReader, [566](#)
  - GetImage, [566](#), [567](#)
  - ImageReader, [566](#)
  - Read, [567](#)
  - ReadACRNEMAImage, [567](#)
  - ReadImage, [567](#)
- gdcm::ImageRegionReader, [568](#)
  - ~ImageRegionReader, [570](#)
  - ComputeBufferLength, [570](#)
  - GetRegion, [571](#)
  - ImageRegionReader, [570](#)
  - Read, [571](#)
  - ReadInformation, [571](#)
  - ReadIntoBuffer, [571](#)
  - SetRegion, [572](#)
- gdcm::ImageToImageFilter, [572](#)
  - ~ImageToImageFilter, [574](#)
  - GetInput, [574](#)
  - GetOutput, [574](#)
  - ImageToImageFilter, [574](#)
- gdcm::ImageWriter, [575](#)

- [~ImageWriter](#), 577
  - [ComputeTargetMediaStorage](#), 577
  - [GetImage](#), 577, 578
  - [ImageWriter](#), 577
  - [Write](#), 578
- [gdcm::ImplicitDataElement](#), 583
  - [GetLength](#), 584
  - [Read](#), 584
  - [ReadPreValue](#), 584
  - [ReadValue](#), 585
  - [ReadValueWithLength](#), 585
  - [ReadWithLength](#), 585
  - [Write](#), 585
- [gdcm::InitializeEvent](#), 586
- [gdcm::Item](#), 601
  - [Clear](#), 603
  - [FindDataElement](#), 603
  - [GetDataElement](#), 603
  - [GetLength](#), 603
  - [GetNestedDataSet](#), 604
  - [InsertDataElement](#), 604
  - [Item](#), 602, 603
  - [operator<<](#), 605
  - [Read](#), 604
  - [SetNestedDataSet](#), 604
  - [Write](#), 605
- [gdcm::IterationEvent](#), 605
- [gdcm::JPEG12Codec](#), 606
  - [~JPEG12Codec](#), 608
  - [DecodeByStreams](#), 608
  - [EncodeBuffer](#), 608
  - [GetHeaderInfo](#), 609
  - [InternalCode](#), 609
  - [IsStateSuspension](#), 609
  - [JPEG12Codec](#), 608
- [gdcm::JPEG16Codec](#), 610
  - [~JPEG16Codec](#), 611
  - [DecodeByStreams](#), 611
  - [EncodeBuffer](#), 612
  - [GetHeaderInfo](#), 612
  - [InternalCode](#), 612
  - [IsStateSuspension](#), 612
  - [JPEG16Codec](#), 611
- [gdcm::JPEG2000Codec](#), 613
  - [~JPEG2000Codec](#), 615
  - [AppendFrameEncode](#), 615
  - [AppendRowEncode](#), 615
  - [Bitmap](#), 619
  - [CanCode](#), 615
  - [CanDecode](#), 616
  - [Clone](#), 616
  - [Code](#), 616
  - [Decode](#), 616
  - [DecodeByStreams](#), 617
  - [DecodeExtent](#), 617
  - [GetHeaderInfo](#), 617
  - [GetQuality](#), 617
  - [GetRate](#), 618
  - [ImageRegionReader](#), 620
  - [IsFrameEncoder](#), 618
  - [IsRowEncoder](#), 618
  - [JPEG2000Codec](#), 615
  - [SetNumberOfResolutions](#), 618
  - [SetQuality](#), 618
  - [SetRate](#), 618
  - [SetReversible](#), 619
  - [SetTileSize](#), 619
  - [StartEncode](#), 619
  - [StopEncode](#), 619
- [gdcm::JPEG8Codec](#), 620
  - [~JPEG8Codec](#), 621
  - [DecodeByStreams](#), 622
  - [EncodeBuffer](#), 622
  - [GetHeaderInfo](#), 622
  - [InternalCode](#), 622
  - [IsStateSuspension](#), 622
  - [JPEG8Codec](#), 621
- [gdcm::JPEGCodec](#), 623
  - [~JPEGCodec](#), 625
  - [AppendFrameEncode](#), 625
  - [AppendRowEncode](#), 626
  - [BitSample](#), 631
  - [CanCode](#), 626
  - [CanDecode](#), 626
  - [Clone](#), 626
  - [Code](#), 627
  - [ComputeOffsetTable](#), 627
  - [Decode](#), 627
  - [DecodeByStreams](#), 627
  - [DecodeExtent](#), 627
  - [EncodeBuffer](#), 628
  - [GetHeaderInfo](#), 628
  - [GetLossless](#), 628
  - [GetQuality](#), 628
  - [ImageRegionReader](#), 631
  - [IsFrameEncoder](#), 629
  - [IsRowEncoder](#), 629
  - [IsStateSuspension](#), 629
  - [IsValid](#), 629
  - [JPEGCodec](#), 625
  - [Quality](#), 631
  - [SetBitSample](#), 629
  - [SetLossless](#), 629
  - [SetPixelFormat](#), 630
  - [SetQuality](#), 630
  - [StartEncode](#), 630
  - [StopEncode](#), 630
- [gdcm::JPEGLSCodec](#), 631

- ~JPEGLSCodec, [634](#)
- AppendFrameEncode, [634](#)
- AppendRowEncode, [634](#)
- CanCode, [634](#)
- CanDecode, [634](#)
- Clone, [635](#)
- Code, [635](#)
- Decode, [635](#)
- DecodeExtent, [636](#)
- GetBufferLength, [636](#)
- GetHeaderInfo, [636](#)
- GetLossless, [636](#)
- ImageRegionReader, [638](#)
- IsFrameEncoder, [637](#)
- IsRowEncoder, [637](#)
- JPEGLSCodec, [633](#)
- SetBufferLength, [637](#)
- SetLossless, [637](#)
- SetLossyError, [637](#)
- StartEncode, [637](#)
- StopEncode, [638](#)
- gdcmm::JSON, [638](#)
  - ~JSON, [639](#)
  - Code, [639](#)
  - Decode, [639](#)
  - GetPrettyPrint, [639](#)
  - JSON, [639](#)
  - PrettyPrintOff, [640](#)
  - PrettyPrintOn, [640](#)
  - SetPrettyPrint, [640](#)
- gdcmm::KAKADUCodec, [641](#)
  - ~KAKADUCodec, [642](#)
  - CanCode, [642](#)
  - CanDecode, [642](#)
  - Clone, [643](#)
  - Code, [643](#)
  - Decode, [643](#)
  - KAKADUCodec, [642](#)
- gdcmm::LO, [644](#)
  - const\_iterator, [645](#)
  - const\_reference, [645](#)
  - const\_reverse\_iterator, [645](#)
  - difference\_type, [645](#)
  - IsValid, [647](#)
  - iterator, [645](#)
  - LO, [647](#)
  - pointer, [646](#)
  - reference, [646](#)
  - reverse\_iterator, [646](#)
  - size\_type, [646](#)
  - Superclass, [646](#)
  - value\_type, [646](#)
- gdcmm::LookupTable, [648](#)
  - ~LookupTable, [650](#)
  - Allocate, [650](#)
  - BitSample, [655](#)
  - Clear, [651](#)
  - Decode, [651](#)
  - GetBitSample, [651](#)
  - GetBufferAsRGBA, [651](#)
  - GetLUTDescriptor, [652](#)
  - GetLUTLength, [652](#)
  - GetLUT, [652](#)
  - GetPointer, [652](#)
  - IncompleteLUT, [655](#)
  - InitializeBlueLUT, [652](#)
  - InitializeGreenLUT, [653](#)
  - InitializeLUT, [653](#)
  - InitializeRedLUT, [653](#)
  - Initialized, [653](#)
  - Internal, [655](#)
  - LookupTable, [650](#)
  - LookupTableType, [650](#)
  - Print, [653](#)
  - SetBlueLUT, [654](#)
  - SetGreenLUT, [654](#)
  - SetLUT, [654](#)
  - SetRedLUT, [654](#)
  - WriteBufferAsRGBA, [654](#)
- gdcmm::MD5, [664](#)
  - Compute, [665](#)
  - ComputeFile, [665](#)
- gdcmm::Macro, [656](#)
  - AddMacroEntry, [658](#)
  - ArrayIncludeMacrosType, [657](#)
  - Clear, [658](#)
  - FindMacroEntry, [658](#)
  - GetMacroEntry, [658](#)
  - GetName, [659](#)
  - Macro, [658](#)
  - MapModuleEntry, [657](#)
  - operator<<, [659](#)
  - SetName, [659](#)
  - Verify, [659](#)
- gdcmm::Macros, [660](#)
  - AddMacro, [661](#)
  - Clear, [661](#)
  - GetMacro, [661](#)
  - IsEmpty, [661](#)
  - Macros, [661](#)
  - ModuleMapType, [660](#)
  - operator<<, [662](#)
- gdcmm::MediaStorage, [665](#)
  - GetMSString, [672](#)
  - GetMSType, [672](#)
  - GetModality, [671](#)
  - GetModalityDimension, [672](#)
  - GetNumberOfMSString, [672](#)

- GetNumberOfMSType, [673](#)
- GetNumberOfModality, [672](#)
- GetString, [673](#)
- GuessFromModality, [673](#)
- IsImage, [673](#)
- IsUndefined, [673](#)
- MSType, [668](#)
- MediaStorage, [671](#)
- ObjectType, [671](#)
- operator MSType, [674](#)
- operator<<, [675](#)
- SetFromDataSet, [674](#)
- SetFromFile, [674](#)
- SetFromHeader, [674](#)
- SetFromModality, [674](#)
- SetFromSourceImageSequence, [675](#)
- gdcmmemberCommand
  - ~MemberCommand, [678](#)
  - Execute, [678](#)
  - m\_ConstMemberFunction, [679](#)
  - m\_MemberFunction, [679](#)
  - m\_This, [680](#)
  - MemberCommand, [678](#)
  - New, [679](#)
  - Self, [677](#)
  - SetCallbackFunction, [679](#)
  - TConstMemberFunctionPointer, [677](#)
  - TMemberFunctionPointer, [678](#)
- gdcmmemberCommand< T >, [675](#)
- gdcmmeshPrimitive, [680](#)
  - ~MeshPrimitive, [683](#)
  - AddPrimitiveData, [683](#)
  - GetMPType, [683](#)
  - GetMPTypeString, [683](#)
  - GetNumberOfPrimitivesData, [684](#)
  - GetPrimitiveData, [684](#)
  - GetPrimitiveType, [685](#)
  - GetPrimitivesData, [684](#)
  - MPType, [682](#)
  - MeshPrimitive, [683](#)
  - PrimitiveData, [685](#)
  - PrimitiveType, [686](#)
  - PrimitivesData, [682](#)
  - SetPrimitiveData, [685](#)
  - SetPrimitiveType, [685](#)
  - SetPrimitivesData, [685](#)
- gdcmmodalityPerformedProcedureStepCreateQuery, [686](#)
  - GetAbstractSyntaxUID, [688](#)
  - GetRequiredDataSet, [688](#)
  - ModalityPerformedProcedureStepCreateQuery, [688](#)
  - QueryFactory, [688](#)
  - ValidateQuery, [688](#)
- gdcmmodalityPerformedProcedureStepSetQuery, [689](#)
  - GetAbstractSyntaxUID, [691](#)
  - GetRequiredDataSet, [691](#)
  - ModalityPerformedProcedureStepSetQuery, [691](#)
  - QueryFactory, [691](#)
  - ValidateQuery, [691](#)
- gdcmmodifiedEvent, [692](#)
- gdcmmodule, [693](#)
  - AddMacro, [694](#)
  - AddModuleEntry, [694](#)
  - ArrayIncludeMacrosType, [694](#)
  - Clear, [694](#)
  - FindModuleEntryInMacros, [695](#)
  - GetModuleEntryInMacros, [695](#)
  - GetName, [695](#)
  - MapModuleEntry, [694](#)
  - Module, [694](#)
  - operator<<, [696](#)
  - SetName, [695](#)
  - Verify, [695](#)
- gdcmmoduleEntry, [696](#)
  - ~ModuleEntry, [698](#)
  - DataElementType, [700](#)
  - Description, [698](#)
  - DescriptionField, [700](#)
  - GetDescription, [699](#)
  - GetName, [699](#)
  - GetType, [699](#)
  - ModuleEntry, [698](#)
  - Name, [700](#)
  - operator<<, [700](#)
  - SetDescription, [699](#)
  - SetName, [699](#)
  - SetType, [699](#)
- gdcmmodules, [701](#)
  - AddModule, [702](#)
  - Clear, [702](#)
  - GetModule, [702](#)
  - IsEmpty, [702](#)
  - ModuleMapType, [701](#)
  - Modules, [702](#)
  - operator<<, [703](#)
- gdcmmovePatientRootQuery, [703](#)
  - GetAbstractSyntaxUID, [705](#)
  - GetTagListByLevel, [705](#)
  - InitializeDataSet, [705](#)
  - MovePatientRootQuery, [704](#)
  - QueryFactory, [706](#)
  - ValidateQuery, [705](#)
- gdcmmoveStudyRootQuery, [706](#)
  - GetAbstractSyntaxUID, [708](#)
  - GetTagListByLevel, [708](#)
  - InitializeDataSet, [708](#)
  - MoveStudyRootQuery, [707](#)
  - QueryFactory, [709](#)



- ValidateQuery, [708](#)
- gdcmm::MrProtocol, [709](#)
  - ~MrProtocol, [710](#)
  - FindMrProtocolByName, [710](#)
  - GetMrProtocolByName, [710](#)
  - GetSliceArray, [710](#)
  - GetVersion, [711](#)
  - Load, [711](#)
  - MrProtocol, [710](#)
  - operator<<, [711](#)
  - Print, [711](#)
- gdcmm::MrProtocol::Slice, [988](#)
  - Normal, [988](#)
  - Position, [988](#)
- gdcmm::MrProtocol::SliceArray, [989](#)
  - Slices, [989](#)
- gdcmm::MrProtocol::Vector3, [1248](#)
  - dCor, [1248](#)
  - dSag, [1248](#)
  - dTra, [1248](#)
- gdcmm::NestedModuleEntries, [721](#)
  - AddModuleEntry, [723](#)
  - GetModuleEntry, [723](#)
  - GetNumberOfModuleEntries, [723](#)
  - NestedModuleEntries, [722](#)
  - operator<<, [723](#)
  - SizeType, [722](#)
- gdcmm::NoEvent, [730](#)
- gdcmm::NormalizedNetworkFunctions, [732](#)
  - ConstructQuery, [733](#)
  - NAction, [733](#)
  - NCreate, [734](#)
  - NDelete, [734](#)
  - NEventReport, [734](#)
  - NGet, [734](#)
  - NSet, [735](#)
- gdcmm::Object, [738](#)
  - ~Object, [740](#)
  - Object, [740](#)
  - operator<<, [741](#)
  - operator=, [741](#)
  - Print, [741](#)
  - Register, [741](#)
  - SmartPointer, [742](#)
  - UnRegister, [741](#)
- gdcmm::OpenSSLCryptoFactory, [742](#)
  - CreateCMSProvider, [743](#)
  - InitOpenSSL, [743](#)
  - OpenSSLCryptoFactory, [743](#)
- gdcmm::OpenSSLCryptographicMessageSyntax, [744](#)
  - ~OpenSSLCryptographicMessageSyntax, [745](#)
  - Decrypt, [745](#)
  - Encrypt, [745](#)
  - GetCipherType, [746](#)
  - OpenSSLCryptographicMessageSyntax, [745](#)
  - ParseCertificateFile, [746](#)
  - ParseKeyFile, [746](#)
  - SetCipherType, [746](#)
  - SetPassword, [746](#)
- gdcmm::OpenSSL7CryptoFactory, [747](#)
  - CreateCMSProvider, [748](#)
  - OpenSSL7CryptoFactory, [748](#)
- gdcmm::OpenSSL7CryptographicMessageSyntax, [749](#)
  - ~OpenSSL7CryptographicMessageSyntax, [750](#)
  - Decrypt, [750](#)
  - Encrypt, [750](#)
  - GetCipherType, [751](#)
  - OpenSSL7CryptographicMessageSyntax, [750](#)
  - ParseCertificateFile, [751](#)
  - ParseKeyFile, [751](#)
  - SetCipherType, [751](#)
  - SetPassword, [751](#)
- gdcmm::Orientation, [752](#)
  - ~Orientation, [753](#)
  - GetLabel, [754](#)
  - GetMajorAxisFromPatientRelativeDirectionCosine, [754](#)
  - GetObliquityThresholdCosineValue, [754](#)
  - GetType, [754](#)
  - operator<<, [755](#)
  - Orientation, [753](#)
  - OrientationType, [753](#)
  - Print, [754](#)
  - SetObliquityThresholdCosineValue, [755](#)
- gdcmm::Overlay, [756](#)
  - ~Overlay, [759](#)
  - Decompress, [759](#)
  - GetBitPosition, [759](#)
  - GetBitsAllocated, [760](#)
  - GetColumns, [760](#)
  - GetDescription, [760](#)
  - GetGroup, [760](#)
  - GetOrigin, [760](#)
  - GetOverlayData, [760](#)
  - GetOverlayTypeAsString, [761](#)
  - GetOverlayTypeFromString, [761](#)
  - GetRows, [761](#)
  - GetType, [761](#)
  - GetTypeAsEnum, [761](#)
  - GetUnpackBuffer, [761](#)
  - GetUnpackBufferLength, [762](#)
  - GrabOverlayFromPixelData, [762](#)
  - IsEmpty, [762](#)
  - IsInPixelData, [762](#)
  - IsZero, [762](#)
  - operator=, [763](#)
  - Overlay, [759](#)
  - OverlayType, [758](#)



- Print, [763](#)
- SetBitPosition, [763](#)
- SetBitsAllocated, [763](#)
- SetColumns, [763](#)
- SetDescription, [764](#)
- SetFrameOrigin, [764](#)
- SetGroup, [764](#)
- SetNumberOfFrames, [764](#)
- SetOrigin, [764](#)
- SetOverlay, [765](#)
- SetRows, [765](#)
- SetType, [765](#)
- Update, [765](#)
- gdcmm::PDBElement, [776](#)
  - GetName, [777](#)
  - GetValue, [778](#)
  - NameField, [779](#)
  - operator<<, [778](#)
  - operator==, [778](#)
  - PDBElement, [777](#)
  - SetName, [778](#)
  - SetValue, [778](#)
  - ValueField, [779](#)
- gdcmm::PDBHeader, [779](#)
  - ~PDBHeader, [780](#)
  - FindPDBElementByName, [781](#)
  - GetPDBEEnd, [781](#)
  - GetPDBElementByName, [781](#)
  - GetPDBInfoTag, [781](#)
  - LoadFromDataElement, [781](#)
  - operator<<, [782](#)
  - PDBHeader, [780](#)
  - Print, [782](#)
- gdcmm::PDFCodec, [783](#)
  - ~PDFCodec, [784](#)
  - CanCode, [784](#)
  - CanDecode, [784](#)
  - Decode, [785](#)
  - PDFCodec, [784](#)
- gdcmm::PGXCodec, [792](#)
  - ~PGXCodec, [793](#)
  - CanCode, [793](#)
  - CanDecode, [793](#)
  - Clone, [794](#)
  - GetHeaderInfo, [794](#)
  - PGXCodec, [793](#)
  - Read, [794](#)
  - Write, [794](#)
- gdcmm::PNMCodec, [825](#)
  - ~PNMCodec, [826](#)
  - CanCode, [826](#)
  - CanDecode, [827](#)
  - Clone, [827](#)
  - GetBufferLength, [827](#)
  - GetHeaderInfo, [827](#)
  - PNMCodec, [826](#)
  - Read, [827](#)
  - SetBufferLength, [828](#)
  - Write, [828](#)
- gdcmm::PVRGCodec, [866](#)
  - ~PVRGCodec, [867](#)
  - CanCode, [867](#)
  - CanDecode, [868](#)
  - Clone, [868](#)
  - Code, [868](#)
  - Decode, [868](#)
  - PVRGCodec, [867](#)
  - SetLossyFlag, [869](#)
- gdcmm::ParseException, [766](#)
  - ~ParseException, [767](#)
  - GetLastElement, [767](#)
  - operator=, [767](#)
  - ParseException, [767](#)
  - SetLastElement, [767](#)
- gdcmm::Parser, [768](#)
  - ~Parser, [770](#)
  - EndElementHandler, [769](#)
  - ErrorType, [769](#)
  - GetBuffer, [770](#)
  - GetCurrentByteIndex, [770](#)
  - GetErrorCode, [770](#)
  - GetErrorString, [771](#)
  - GetUserData, [771](#)
  - Parse, [771](#)
  - ParseBuffer, [771](#)
  - Parser, [770](#)
  - Process, [771](#)
  - SetElementHandler, [771](#)
  - SetUserData, [772](#)
  - StartElementHandler, [769](#)
- gdcmm::Patient, [772](#)
  - Patient, [772](#)
- gdcmm::PersonName, [789](#)
  - Component, [791](#)
  - GetMaxLength, [789](#)
  - GetNumberOfComponents, [790](#)
  - MaxLength, [791](#)
  - MaxNumberOfComponents, [791](#)
  - Padding, [791](#)
  - Print, [790](#)
  - Separator, [791](#)
  - SetBlob, [790](#)
  - SetComponents, [790](#)
- gdcmm::PhotometricInterpretation, [795](#)
  - GetPIString, [797](#)
  - GetPIType, [797](#)
  - GetSamplesPerPixel, [797](#)
  - GetString, [797](#)

- GetType, 797
- IsLossless, 797
- IsLossy, 797
- IsRetired, 798
- IsSameColorSpace, 798
- operator PType, 798
- operator < <, 798
- PType, 796
- PhotometricInterpretation, 796
- gdcmm::PixelFormat, 798
  - Bitmap, 807
  - GetBitsAllocated, 801
  - GetBitsStored, 802
  - GetHighBit, 802
  - GetMax, 802
  - GetMin, 802
  - GetPixelRepresentation, 802
  - GetPixelSize, 803
  - GetSamplesPerPixel, 803
  - GetScalarType, 803
  - GetScalarTypeAsString, 803
  - IsCompatible, 804
  - IsValid, 804
  - operator ScalarType, 804
  - operator !=, 804
  - operator < <, 807
  - operator ==, 804, 805
  - PixelFormat, 801
  - Print, 805
  - ScalarType, 800
  - SetBitsAllocated, 805
  - SetBitsStored, 805
  - SetHighBit, 805
  - SetPixelRepresentation, 805
  - SetSamplesPerPixel, 806
  - SetScalarType, 806
  - Validate, 806
- gdcmm::Pixmap, 807
  - ~Pixmap, 809
  - AreOverlaysInPixelData, 809
  - Curves, 812
  - GetCurve, 809, 810
  - GetIconImage, 810
  - GetNumberOfCurves, 810
  - GetNumberOfOverlays, 810
  - GetOverlay, 810, 811
  - Icon, 812
  - Overlays, 812
  - Pixmap, 809
  - Print, 811
  - RemoveOverlay, 811
  - SetIconImage, 811
  - SetNumberOfCurves, 811
  - SetNumberOfOverlays, 811
  - UnusedBitsPresentInPixelData, 812
- gdcmm::PixmapReader, 813
  - ~PixmapReader, 815
  - GetPixmap, 815
  - PixelData, 817
  - PixmapReader, 815
  - Read, 816
  - ReadACRNEMAIImage, 816
  - ReadImage, 816
  - ReadImageInternal, 816
- gdcmm::PixmapToPixmapFilter, 817
  - ~PixmapToPixmapFilter, 819
  - GetInput, 819
  - GetOutput, 819
  - GetOutputAsPixmap, 819
  - PixmapToPixmapFilter, 819
- gdcmm::PixmapWriter, 820
  - ~PixmapWriter, 822
  - DolconImage, 822
  - GetImage, 823
  - GetPixmap, 823
  - PixelData, 824
  - PixmapWriter, 822
  - PrepareWrite, 823
  - SetImage, 823
  - SetPixmap, 824
  - Write, 824
- gdcmm::Preamble, 828
  - ~Preamble, 830
  - Clear, 830
  - Create, 830
  - GetInternal, 830
  - GetLength, 830
  - IsEmpty, 831
  - IsValid, 831
  - operator < <, 832
  - operator =, 831
  - Preamble, 829, 830
  - Print, 831
  - Read, 831
  - Remove, 831
  - Valid, 832
  - Write, 832
- gdcmm::PresentationContext, 833
  - AbstractSyntax, 836
  - AddTransferSyntax, 835
  - GetAbstractSyntax, 835
  - GetNumberOfTransferSyntaxes, 835
  - GetPresentationContextID, 835
  - GetTransferSyntax, 835
  - ID, 836
  - operator ==, 835
  - PresentationContext, 834
  - Print, 836

- SetAbstractSyntax, [836](#)
- SetPresentationContextID, [836](#)
- SizeType, [834](#)
- TransferSyntaxArrayType, [834](#)
- TransferSyntaxes, [836](#)
- gdcmm::PresentationContextGenerator, [839](#)
  - AddFromFile, [841](#)
  - AddPresentationContext, [841](#)
  - GenerateFromFilenames, [841](#)
  - GenerateFromUID, [842](#)
  - GetDefaultTransferSyntax, [842](#)
  - GetPresentationContexts, [842](#)
  - PresentationContextArrayType, [841](#)
  - PresentationContextGenerator, [841](#)
  - SetDefaultTransferSyntax, [842](#)
  - SetMergeModeToAbstractSyntax, [842](#)
  - SetMergeModeToTransferSyntax, [843](#)
  - SizeType, [841](#)
- gdcmm::Printer, [851](#)
  - ~Printer, [853](#)
  - F, [856](#)
  - GetPrintStyle, [854](#)
  - MaxPrintLength, [856](#)
  - Print, [854](#)
  - PrintDataElement, [854](#)
  - PrintDataSet, [854](#)
  - PrintSQ, [855](#)
  - PrintStyle, [856](#)
  - PrintStyles, [853](#)
  - Printer, [853](#)
  - SetColor, [855](#)
  - SetFile, [855](#)
  - SetStyle, [855](#)
- gdcmm::PrivateDict, [856](#)
  - ~PrivateDict, [857](#)
  - AddDictEntry, [857](#)
  - Dicts, [859](#)
  - FindDictEntry, [858](#)
  - GetDictEntry, [858](#)
  - IsEmpty, [858](#)
  - LoadDefault, [858](#)
  - operator<<, [859](#)
  - PrintXML, [858](#)
  - PrivateDict, [857](#)
  - RemoveDictEntry, [858](#)
- gdcmm::PrivateTag, [859](#)
  - GetAsDataElement, [861](#)
  - GetOwner, [861](#)
  - operator<, [861](#)
  - operator<<, [862](#)
  - PrivateTag, [860](#), [861](#)
  - ReadFromCommaSeparatedString, [861](#)
  - SetOwner, [862](#)
- gdcmm::ProgressEvent, [862](#)
  - ~ProgressEvent, [864](#)
  - CheckEvent, [864](#)
  - GetEventName, [865](#)
  - GetProgress, [865](#)
  - MakeObject, [865](#)
  - ProgressEvent, [864](#)
  - Self, [864](#)
  - SetProgress, [865](#)
  - Superclass, [864](#)
- gdcmm::PythonFilter, [869](#)
  - ~PythonFilter, [870](#)
  - GetFile, [870](#)
  - PythonFilter, [870](#)
  - SetDicts, [870](#)
  - SetFile, [870](#)
  - ToPyObject, [870](#)
  - UseDictAlways, [871](#)
- gdcmm::QueryBase, [871](#)
  - ~QueryBase, [872](#)
  - GetAllRequiredTags, [872](#)
  - GetAllTags, [872](#)
  - GetHierarchicalSearchTags, [873](#)
  - GetName, [873](#)
  - GetOptionalTags, [873](#)
  - GetQueryLevel, [873](#)
  - GetRequiredTags, [873](#)
  - GetUniqueTags, [874](#)
- gdcmm::QueryFactory, [874](#)
  - GetCharacterFromCurrentLocale, [875](#)
  - ListCharSets, [875](#)
  - ProduceCharacterSetDataElement, [875](#)
  - ProduceQuery, [875](#)
- gdcmm::QueryImage, [876](#)
  - GetHierarchicalSearchTags, [877](#)
  - GetName, [877](#)
  - GetOptionalTags, [877](#)
  - GetQueryLevel, [877](#)
  - GetRequiredTags, [878](#)
  - GetUniqueTags, [878](#)
- gdcmm::QueryPatient, [878](#)
  - GetHierarchicalSearchTags, [879](#)
  - GetName, [879](#)
  - GetOptionalTags, [880](#)
  - GetQueryLevel, [880](#)
  - GetRequiredTags, [880](#)
  - GetUniqueTags, [880](#)
- gdcmm::QuerySeries, [881](#)
  - GetHierarchicalSearchTags, [882](#)
  - GetName, [882](#)
  - GetOptionalTags, [882](#)
  - GetQueryLevel, [882](#)
  - GetRequiredTags, [882](#)
  - GetUniqueTags, [883](#)
- gdcmm::QueryStudy, [883](#)

- GetHierarchicalSearchTags, 884
- GetName, 884
- GetOptionalTags, 885
- GetQueryLevel, 885
- GetRequiredTags, 885
- GetUniqueTags, 885
- gdcm::RAWCodec, 886
  - ~RAWCodec, 887
  - CanCode, 887
  - CanDecode, 888
  - Clone, 888
  - Code, 888
  - Decode, 888
  - DecodeByStreams, 889
  - DecodeBytes, 889
  - GetHeaderInfo, 889
  - RAWCodec, 887
- gdcm::RLECodec, 906
  - ~RLECodec, 908
  - AppendFrameEncode, 908
  - AppendRowEncode, 908
  - CanCode, 909
  - CanDecode, 909
  - Clone, 909
  - Code, 909
  - Decode, 909
  - DecodeByStreams, 910
  - DecodeExtent, 910
  - GetBufferLength, 910
  - GetHeaderInfo, 910
  - ImageRegionReader, 912
  - IsFrameEncoder, 910
  - IsRowEncoder, 911
  - RLECodec, 908
  - SetBufferLength, 911
  - SetLength, 911
  - StartEncode, 911
  - StopEncode, 911
- gdcm::Reader, 890
  - ~Reader, 893
  - CanRead, 893
  - F, 897
  - GetFile, 893
  - GetStreamCurrentPosition, 894
  - GetStreamPtr, 894
  - Read, 894
  - ReadDataSet, 894
  - ReadMetaInformation, 895
  - ReadPreamble, 895
  - ReadSelectedPrivateTags, 895
  - ReadSelectedTags, 895
  - ReadUpToTag, 895
  - Reader, 893
  - SetFile, 896
  - SetFileName, 896
  - SetStream, 896
  - StreamImageReader, 897
- gdcm::RealWorldValueMappingContent, 897
  - CodeMeaning, 898
  - CodeValue, 898
  - RealWorldValueIntercept, 898
  - RealWorldValueSlope, 898
- gdcm::Region, 899
  - ~Region, 900
  - Area, 900
  - Clone, 900
  - ComputeBoundingBox, 900
  - Empty, 900
  - IsValid, 901
  - Print, 901
  - Region, 899
- gdcm::Rescaler, 901
  - ~Rescaler, 903
  - ComputeInterceptSlopePixelType, 903
  - ComputePixelTypeFromMinMax, 903
  - GetIntercept, 904
  - GetSlope, 904
  - InverseRescale, 904
  - InverseRescaleFunctionIntoBestFit, 904
  - Rescale, 904
  - RescaleFunctionIntoBestFit, 904
  - Rescaler, 903
  - SetIntercept, 905
  - SetMinMaxForPixelType, 905
  - SetPixelFormat, 905
  - SetSlope, 905
  - SetTargetPixelType, 905
  - SetUseTargetPixelType, 906
- gdcm::SHA1, 978
  - ~SHA1, 979
  - Compute, 979
  - ComputeFile, 979
  - SHA1, 979
- gdcm::SOPClassUIDToIOD, 996
  - const, 996
  - GetIODFromSOPClassUID, 997
  - GetIOD, 997
  - GetNumberOfSOPClassToIOD, 997
  - GetSOPClassUIDFromIOD, 997
  - GetSOPClassUIDToIODs, 997
  - GetSOPClassUIDToIOD, 997
- gdcm::STATIC\_ASSERTION\_FAILURE< true >, 1011
- gdcm::STATIC\_ASSERTION\_FAILURE< x >, 1010
- gdcm::Scanner, 915
  - ~Scanner, 919
  - AddPrivateTag, 919
  - AddSkipTag, 919
  - AddTag, 919

- Begin, [920](#)
- ClearSkipTags, [920](#)
- ClearTags, [920](#)
- ConstIterator, [918](#)
- End, [920](#)
- GetAllFileNamesFromTagToValue, [920](#)
- GetFilenameFromTagToValue, [920](#)
- GetFileNames, [921](#)
- GetKeys, [921](#)
- GetMapping, [921](#)
- GetMappingFromTagToValue, [921](#)
- GetMappings, [921](#)
- GetOrderedValues, [922](#)
- GetValue, [922](#)
- GetValues, [922](#)
- IsKey, [923](#)
- MappingType, [918](#)
- New, [923](#)
- operator<<, [924](#)
- Print, [923](#)
- ProcessPublicTag, [923](#)
- Scan, [924](#)
- Scanner, [919](#)
- TagToValue, [918](#)
- TagToValueValueType, [918](#)
- ValueType, [919](#)
- gdcmm::Scanner::Itstr, [655](#)
- operator(), [656](#)
- gdcmm::Segment, [925](#)
- ~Segment, [928](#)
- ALGOType, [927](#)
- AddSurface, [928](#)
- AnatomicRegion, [933](#)
- AnatomicRegionModifiers, [933](#)
- BasicCodedEntryVector, [927](#)
- GetALGOType, [928](#)
- GetALGOTypeString, [928](#)
- GetAnatomicRegion, [929](#)
- GetAnatomicRegionModifiers, [929](#)
- GetPropertyCategory, [929](#)
- GetPropertyType, [929](#), [930](#)
- GetPropertyTypeModifiers, [930](#)
- GetSegmentAlgorithmName, [930](#)
- GetSegmentAlgorithmType, [930](#)
- GetSegmentDescription, [930](#)
- GetSegmentLabel, [930](#)
- GetSegmentNumber, [931](#)
- GetSurface, [931](#)
- GetSurfaceCount, [931](#)
- GetSurfaces, [931](#)
- PropertyCategory, [934](#)
- PropertyType, [934](#)
- PropertyTypeModifiers, [934](#)
- Segment, [928](#)
- SegmentAlgorithmName, [934](#)
- SegmentAlgorithmType, [934](#)
- SegmentDescription, [934](#)
- SegmentLabel, [934](#)
- SegmentNumber, [935](#)
- SetAnatomicRegion, [931](#)
- SetAnatomicRegionModifiers, [931](#)
- SetPropertyCategory, [932](#)
- SetPropertyType, [932](#)
- SetPropertyTypeModifiers, [932](#)
- SetSegmentAlgorithmName, [932](#)
- SetSegmentAlgorithmType, [932](#)
- SetSegmentDescription, [933](#)
- SetSegmentLabel, [933](#)
- SetSegmentNumber, [933](#)
- SetSurfaceCount, [933](#)
- SurfaceCount, [935](#)
- SurfaceVector, [927](#)
- Surfaces, [935](#)
- gdcmm::SegmentHelper, [84](#)
- gdcmm::SegmentHelper::BasicCodedEntry, [187](#)
- BasicCodedEntry, [188](#)
- CSD, [189](#)
- CSV, [189](#)
- CM, [189](#)
- CV, [189](#)
- IsEmpty, [188](#)
- gdcmm::SegmentReader, [938](#)
- ~SegmentReader, [940](#)
- GetSegments, [940](#)
- Read, [940](#)
- ReadSegment, [941](#)
- ReadSegments, [941](#)
- SegmentMap, [939](#)
- SegmentReader, [940](#)
- SegmentVector, [940](#)
- Segments, [941](#)
- gdcmm::SegmentWriter, [942](#)
- ~SegmentWriter, [943](#)
- AddSegment, [944](#)
- GetNumberOfSegments, [944](#)
- GetSegment, [944](#)
- GetSegments, [944](#)
- PrepareWrite, [944](#)
- SegmentVector, [943](#)
- SegmentWriter, [943](#)
- Segments, [945](#)
- SetNumberOfSegments, [944](#)
- SetSegments, [945](#)
- Write, [945](#)
- gdcmm::SegmentedPaletteColorLookupTable, [935](#)
- ~SegmentedPaletteColorLookupTable, [936](#)
- Print, [937](#)
- SegmentedPaletteColorLookupTable, [936](#)

- SetLUT, [937](#)
- gdcm::SequenceOfFragments, [946](#)
  - AddFragment, [948](#)
  - Begin, [949](#)
  - Clear, [949](#)
  - ComputeByteLength, [949](#)
  - ComputeLength, [949](#)
  - ConstIterator, [948](#)
  - End, [949](#), [950](#)
  - FragmentVector, [948](#)
  - GetBuffer, [950](#)
  - GetFragBuffer, [950](#)
  - GetFragment, [950](#)
  - GetLength, [950](#)
  - GetNumberOfFragments, [951](#)
  - GetTable, [951](#)
  - Iterator, [948](#)
  - New, [951](#)
  - operator==, [951](#)
  - Print, [951](#)
  - Read, [952](#)
  - ReadPreValue, [952](#)
  - ReadValue, [952](#)
  - SequenceOfFragments, [948](#)
  - SetLength, [952](#)
  - SizeType, [948](#)
  - Write, [952](#)
  - WriteBuffer, [953](#)
- gdcm::SequenceOfItems, [953](#)
  - AddItem, [957](#)
  - AddNewUndefinedLengthItem, [957](#)
  - Begin, [957](#)
  - Clear, [957](#)
  - ComputeLength, [958](#)
  - ConstIterator, [956](#)
  - End, [958](#)
  - FindDataElement, [958](#)
  - GetItem, [958](#)
  - GetLength, [959](#)
  - GetNumberOfItems, [959](#)
  - IsEmpty, [959](#)
  - IsUndefinedLength, [959](#)
  - ItemVector, [956](#)
  - Items, [962](#)
  - Iterator, [956](#)
  - New, [959](#)
  - operator=, [960](#)
  - operator==, [960](#)
  - Print, [960](#)
  - Read, [960](#)
  - RemoveItemByIndex, [961](#)
  - SequenceLengthField, [962](#)
  - SequenceOfItems, [956](#)
  - SetLength, [961](#)
  - SetLengthToUndefined, [961](#)
  - SetNumberOfItems, [961](#)
  - SizeType, [956](#)
  - Write, [962](#)
- gdcm::SerieHelper, [963](#)
  - ~SerieHelper, [965](#)
  - AddFile, [965](#)
  - AddFileName, [965](#)
  - AddRestriction, [965](#)
  - Clear, [966](#)
  - CreateDefaultUniqueSeriesIdentifier, [966](#)
  - CreateUniqueSeriesIdentifier, [966](#)
  - FileNameOrdering, [966](#)
  - GetFirstSingleSerieUIDFileSet, [966](#)
  - GetNextSingleSerieUIDFileSet, [966](#)
  - ImagePositionPatientOrdering, [966](#)
  - ItFileSetHt, [967](#)
  - OrderFileList, [967](#)
  - SerieHelper, [964](#)
  - SerieRestrictions, [964](#)
  - SetDirectory, [967](#)
  - SetLoadMode, [967](#)
  - SetUseSeriesDetails, [967](#)
  - SingleSerieUIDFileSetHT, [968](#)
  - SingleSerieUIDFileSetmap, [964](#)
  - UserOrdering, [967](#)
- gdcm::SerieHelper::Rule, [914](#)
  - elem, [914](#)
  - group, [914](#)
  - op, [915](#)
  - value, [915](#)
- gdcm::Series, [968](#)
  - Series, [968](#)
- gdcm::ServiceClassUser, [970](#)
  - ~ServiceClassUser, [972](#)
  - GetAETitle, [973](#)
  - GetCalledAETitle, [973](#)
  - GetTimeout, [973](#)
  - InitializeConnection, [973](#)
  - IsPresentationContextAccepted, [973](#)
  - New, [974](#)
  - SendEcho, [974](#)
  - SendFind, [974](#)
  - SendMove, [974](#), [975](#)
  - SendStore, [975](#)
  - ServiceClassUser, [972](#)
  - SetAETitle, [975](#)
  - SetCalledAETitle, [976](#)
  - SetHostname, [976](#)
  - SetPort, [976](#)
  - SetPortSCP, [976](#)
  - SetPresentationContexts, [977](#)
  - SetTimeout, [977](#)
  - StartAssociation, [977](#)

- StopAssociation, [977](#)
- gdcmm::SimpleMemberCommand
  - ~SimpleMemberCommand, [982](#)
  - Execute, [983](#)
  - m\_MemberFunction, [984](#)
  - m\_This, [984](#)
  - New, [983](#)
  - Self, [982](#)
  - SetCallbackFunction, [983](#)
  - SimpleMemberCommand, [982](#)
  - TMemberFunctionPointer, [982](#)
- gdcmm::SimpleMemberCommand< T >, [980](#)
- gdcmm::SimpleSubjectWatcher, [984](#)
  - ~SimpleSubjectWatcher, [985](#)
  - EndFilter, [985](#)
  - ShowAbort, [986](#)
  - ShowAnonymization, [986](#)
  - ShowData, [986](#)
  - ShowDataSet, [986](#)
  - ShowFileName, [986](#)
  - ShowIteration, [986](#)
  - ShowProgress, [987](#)
  - SimpleSubjectWatcher, [985](#)
  - StartFilter, [987](#)
  - TestAbortOff, [987](#)
  - TestAbortOn, [987](#)
- gdcmm::SmartPointer
  - ~SmartPointer, [992](#)
  - GetPointer, [992](#)
  - operator ObjectType \*, [993](#)
  - operator\*, [993](#)
  - operator->, [993](#)
  - operator=, [993](#), [994](#)
  - SmartPointer, [991](#), [992](#)
- gdcmm::SmartPointer< ObjectType >, [990](#)
- gdcmm::Sorter, [998](#)
  - ~Sorter, [1000](#)
  - AddSelect, [1000](#)
  - FileNames, [1002](#)
  - GetFileNames, [1000](#)
  - operator<<, [1002](#)
  - Print, [1001](#)
  - Selection, [1002](#)
  - SelectionMap, [1000](#)
  - SetSortFunction, [1001](#)
  - SetTagsToRead, [1001](#)
  - Sort, [1001](#)
  - SortFunc, [1003](#)
  - SortFunction, [1000](#)
  - Sorter, [1000](#)
  - StableSort, [1002](#)
  - TagsToRead, [1003](#)
- gdcmm::Spacing, [1003](#)
  - ~Spacing, [1005](#)
  - ComputePixelAspectRatioFromPixelSpacing, [1005](#)
  - Spacing, [1005](#)
  - SpacingType, [1004](#)
- gdcmm::Spectroscopy, [1005](#)
  - Spectroscopy, [1006](#)
- gdcmm::SplitMosaicFilter, [1006](#)
  - ~SplitMosaicFilter, [1007](#)
  - ComputeMOSAICDimensions, [1007](#)
  - ComputeMOSAICSliceNormal, [1007](#)
  - ComputeMOSAICSlicePosition, [1007](#)
  - GetFile, [1008](#)
  - GetImage, [1008](#)
  - SetFile, [1008](#)
  - SetImage, [1008](#)
  - Split, [1009](#)
  - SplitMosaicFilter, [1007](#)
- gdcmm::StartEvent, [1009](#)
- gdcmm::StreamImageReader, [1011](#)
  - ~StreamImageReader, [1012](#)
  - CanReadImage, [1012](#)
  - DefinePixelExtent, [1012](#)
  - DefineProperBufferLength, [1013](#)
  - GetDimensionsValueForResolution, [1013](#)
  - GetFile, [1013](#)
  - Read, [1013](#)
  - ReadImageInformation, [1014](#)
  - SetFileName, [1014](#)
  - SetStream, [1014](#)
  - StreamImageReader, [1012](#)
- gdcmm::StreamImageWriter, [1015](#)
  - ~StreamImageWriter, [1018](#)
  - CanWriteFile, [1018](#)
  - DefinePixelExtent, [1018](#)
  - DefineProperBufferLength, [1018](#)
  - mElementOffsets, [1021](#)
  - mElementOffsets1, [1021](#)
  - mWriter, [1021](#)
  - mXMax, [1021](#)
  - mXMin, [1022](#)
  - mYMax, [1022](#)
  - mYMin, [1022](#)
  - mZMax, [1022](#)
  - mZMin, [1022](#)
  - mspFile, [1021](#)
  - SetFile, [1019](#)
  - SetFileName, [1019](#)
  - SetStream, [1019](#)
  - StreamImageWriter, [1017](#)
  - Write, [1019](#)
  - WriteImageInformation, [1020](#)
  - WriteImageSubregionRAW, [1020](#)
  - WriteRawHeader, [1020](#)
- gdcmm::StrictScanner, [1023](#)
  - ~StrictScanner, [1026](#)



- AddPrivateTag, [1026](#)
- AddSkipTag, [1027](#)
- AddTag, [1027](#)
- Begin, [1027](#)
- ClearSkipTags, [1027](#)
- ClearTags, [1027](#)
- ConstIterator, [1025](#)
- End, [1027](#)
- GetAllFileNamesFromTagToValue, [1028](#)
- GetFilenameFromTagToValue, [1028](#)
- GetFileNames, [1028](#)
- GetKeys, [1028](#)
- GetMapping, [1028](#)
- GetMappingFromTagToValue, [1028](#)
- GetMappings, [1029](#)
- GetOrderedValues, [1029](#)
- GetValue, [1029](#)
- GetValues, [1029](#)
- IsKey, [1030](#)
- MappingType, [1025](#)
- New, [1030](#)
- operator<<, [1031](#)
- Print, [1030](#)
- ProcessPublicTag, [1030](#)
- Scan, [1031](#)
- StrictScanner, [1026](#)
- TagToValue, [1026](#)
- TagToValueValueType, [1026](#)
- ValueType, [1026](#)
- gdcmm::StrictScanner::Itstr, [656](#)
- operator(), [656](#)
- gdcmm::String
  - const\_iterator, [1034](#)
  - const\_reference, [1034](#)
  - const\_reverse\_iterator, [1034](#)
  - difference\_type, [1034](#)
  - IsValid, [1036](#)
  - iterator, [1034](#)
  - operator const char \*, [1036](#)
  - pointer, [1034](#)
  - reference, [1035](#)
  - reverse\_iterator, [1035](#)
  - size\_type, [1035](#)
  - String, [1035](#), [1036](#)
  - Trim, [1036](#), [1037](#)
  - Truncate, [1037](#)
  - value\_type, [1035](#)
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [1032](#)
- gdcmm::StringFilter, [1037](#)
- ~StringFilter, [1038](#)
- ExecuteQuery, [1038](#)
- FromString, [1039](#)
- GetFile, [1039](#)
- SetDicts, [1039](#)
- SetFile, [1039](#)
- StringFilter, [1038](#)
- ToString, [1040](#)
- ToStringPair, [1040](#), [1041](#)
- UseDictAlways, [1041](#)
- gdcmm::Study, [1041](#)
- Study, [1041](#)
- gdcmm::Subject, [1042](#)
- ~Subject, [1043](#)
- AddObserver, [1043](#), [1044](#)
- GetCommand, [1044](#)
- HasObserver, [1044](#)
- InvokeEvent, [1044](#)
- RemoveAllObservers, [1045](#)
- RemoveObserver, [1045](#)
- Subject, [1043](#)
- gdcmm::Surface, [1045](#)
- ~Surface, [1049](#)
- GetAlgorithmFamily, [1049](#)
- GetAlgorithmName, [1049](#)
- GetAlgorithmVersion, [1049](#)
- GetAxisOfRotation, [1050](#)
- GetCenterOfRotation, [1050](#)
- GetFiniteVolume, [1050](#)
- GetManifold, [1050](#)
- GetMaximumPointDistance, [1050](#)
- GetMeanPointDistance, [1050](#)
- GetMeshPrimitive, [1051](#)
- GetNumberOfSurfacePoints, [1051](#)
- GetNumberOfVectors, [1051](#)
- GetPointCoordinatesData, [1051](#)
- GetPointPositionAccuracy, [1051](#)
- GetPointsBoundingBoxCoordinates, [1052](#)
- GetProcessingAlgorithm, [1052](#)
- GetRecommendedDisplayCIELabValue, [1052](#)
- GetRecommendedDisplayGrayscaleValue, [1052](#)
- GetRecommendedPresentationOpacity, [1053](#)
- GetRecommendedPresentationType, [1053](#)
- GetSTATESString, [1053](#)
- GetSTATES, [1053](#)
- GetSurfaceComments, [1053](#)
- GetSurfaceNumber, [1053](#)
- GetSurfaceProcessing, [1053](#)
- GetSurfaceProcessingDescription, [1054](#)
- GetSurfaceProcessingRatio, [1054](#)
- GetVIEWType, [1054](#)
- GetVIEWTypeString, [1055](#)
- GetVectorAccuracy, [1054](#)
- GetVectorCoordinateData, [1054](#)
- GetVectorDimensionality, [1054](#)
- STATES, [1048](#)
- SetAlgorithmFamily, [1055](#)
- SetAlgorithmName, [1055](#)



- SetAlgorithmVersion, [1055](#)
- SetAxisOfRotation, [1055](#)
- SetCenterOfRotation, [1055](#)
- SetFiniteVolume, [1056](#)
- SetManifold, [1056](#)
- SetMaximumPointDistance, [1056](#)
- SetMeanPointDistance, [1056](#)
- SetMeshPrimitive, [1056](#)
- SetNumberOfSurfacePoints, [1056](#)
- SetNumberOfVectors, [1057](#)
- SetPointCoordinatesData, [1057](#)
- SetPointPositionAccuracy, [1057](#)
- SetPointsBoundingBoxCoordinates, [1057](#)
- SetProcessingAlgorithm, [1057](#)
- SetRecommendedDisplayCIELabValue, [1057](#), [1058](#)
- SetRecommendedDisplayGrayscaleValue, [1058](#)
- SetRecommendedPresentationOpacity, [1058](#)
- SetRecommendedPresentationType, [1058](#)
- SetSurfaceComments, [1058](#)
- SetSurfaceNumber, [1059](#)
- SetSurfaceProcessing, [1059](#)
- SetSurfaceProcessingDescription, [1059](#)
- SetSurfaceProcessingRatio, [1059](#)
- SetVectorAccuracy, [1059](#)
- SetVectorCoordinateData, [1059](#)
- SetVectorDimensionality, [1060](#)
- Surface, [1049](#)
- VIEWType, [1048](#)
- gdcm::SurfaceHelper, [1060](#)
  - ColorArray, [1061](#)
  - RGBToRecommendedDisplayCIELab, [1062](#)
  - RGBToRecommendedDisplayGrayscale, [1063](#)
  - RecommendedDisplayCIELabToRGB, [1061](#)
- gdcm::SurfaceReader, [1063](#)
  - ~SurfaceReader, [1065](#)
  - GetNumberOfSurfaces, [1065](#)
  - Read, [1066](#)
  - ReadPointMacro, [1066](#)
  - ReadSurface, [1066](#)
  - ReadSurfaces, [1066](#)
  - SurfaceReader, [1065](#)
- gdcm::SurfaceWriter, [1067](#)
  - ~SurfaceWriter, [1068](#)
  - ComputeNumberOfSurfaces, [1068](#)
  - GetNumberOfSurfaces, [1068](#)
  - NumberOfSurfaces, [1069](#)
  - PrepareWrite, [1069](#)
  - PrepareWritePointMacro, [1069](#)
  - SetNumberOfSurfaces, [1069](#)
  - SurfaceWriter, [1068](#)
  - Write, [1069](#)
- gdcm::SwapCode, [1070](#)
  - GetIndex, [1071](#)
  - GetSwapCodeString, [1072](#)
  - operator SwapCode::SwapCodeType, [1072](#)
  - operator<<, [1072](#)
  - SwapCode, [1071](#)
  - SwapCodeType, [1071](#)
- gdcm::SwapperDoOp, [1072](#)
  - Swap, [1073](#)
  - SwapArray, [1073](#)
- gdcm::SwapperNoOp, [1073](#)
  - Swap, [1073](#)
  - SwapArray, [1074](#)
- gdcm::System, [1074](#)
  - DeleteDirectory, [1075](#)
  - EncodeBytes, [1075](#)
  - FileExists, [1076](#)
  - FileIsDirectory, [1076](#)
  - FileIsSymlink, [1076](#)
  - FileSize, [1076](#)
  - FileTime, [1077](#)
  - FormatDateTime, [1077](#)
  - GetCWD, [1078](#)
  - GetCurrentDateTime, [1077](#)
  - GetCurrentModuleFileName, [1077](#)
  - GetCurrentProcessFileName, [1078](#)
  - GetCurrentResourcesDirectory, [1078](#)
  - GetHostName, [1078](#)
  - GetLastSystemError, [1078](#)
  - GetLocaleCharset, [1078](#)
  - GetPermissions, [1079](#)
  - GetTimezoneOffsetFromUTC, [1079](#)
  - MakeDirectory, [1079](#)
  - ParseDateTime, [1079](#)
  - RemoveFile, [1080](#)
  - SetPermissions, [1080](#)
  - StrCaseCmp, [1080](#)
  - StrNCaseCmp, [1080](#)
  - StrSep, [1081](#)
  - StrTokR, [1081](#)
- gdcm::Table, [1081](#)
  - ~Table, [1082](#)
  - GetTableEntry, [1083](#)
  - InsertEntry, [1083](#)
  - MapTableEntry, [1082](#)
  - operator<<, [1083](#)
  - Table, [1082](#)
- gdcm::TableEntry, [1083](#)
  - ~TableEntry, [1084](#)
  - TableEntry, [1084](#)
- gdcm::TableReader, [1084](#)
  - ~TableReader, [1085](#)
  - CharacterDataHandler, [1086](#)
  - EndElement, [1086](#)
  - GetDefs, [1086](#)
  - GetFilename, [1086](#)
  - HandleIODEntry, [1086](#)

- HandleIOD, [1086](#)
- HandleMacro, [1087](#)
- HandleMacroEntry, [1087](#)
- HandleMacroEntryDescription, [1087](#)
- HandleModule, [1087](#)
- HandleModuleEntry, [1087](#)
- HandleModuleEntryDescription, [1087](#)
- HandleModuleInclude, [1088](#)
- Read, [1088](#)
- SetFilename, [1088](#)
- StartElement, [1088](#)
- TableReader, [1085](#)
- gdcmm::Tag, [1090](#)
  - bytes, [1101](#)
  - GetElement, [1093](#)
  - GetElementTag, [1093](#)
  - GetGroup, [1094](#)
  - GetLength, [1094](#)
  - GetPrivateCreator, [1094](#)
  - IsGroupLength, [1094](#)
  - IsGroupXX, [1095](#)
  - IsIllegal, [1095](#)
  - IsPrivate, [1095](#)
  - IsPrivateCreator, [1095](#)
  - IsPublic, [1096](#)
  - operator!=, [1096](#)
  - operator<, [1096](#)
  - operator<<, [1100](#)
  - operator<=, [1096](#)
  - operator>>, [1100](#)
  - operator=, [1096](#)
  - operator==, [1097](#)
  - operator[], [1097](#)
  - PrintAsContinuousString, [1097](#)
  - PrintAsContinuousUpperCaseString, [1097](#)
  - PrintAsPipeSeparatedString, [1098](#)
  - Read, [1098](#)
  - ReadFromCommaSeparatedString, [1098](#)
  - ReadFromContinuousString, [1098](#)
  - ReadFromPipeSeparatedString, [1098](#)
  - SetElement, [1099](#)
  - SetElementTag, [1099](#)
  - SetGroup, [1099](#)
  - SetPrivateCreator, [1100](#)
  - Tag, [1092](#), [1093](#)
  - tag, [1101](#)
  - tags, [1101](#)
  - Write, [1100](#)
- gdcmm::TagPath, [1101](#)
  - ~TagPath, [1102](#)
  - ConstructFromString, [1102](#)
  - ConstructFromTagList, [1103](#)
  - IsValid, [1103](#)
  - Print, [1103](#)
  - Push, [1103](#)
  - TagPath, [1102](#)
- gdcmm::Testing, [1104](#)
  - ~Testing, [1106](#)
  - ComputeFileMD5, [1106](#)
  - ComputeMD5, [1106](#)
  - GetDataExtraRoot, [1106](#)
  - GetDataRoot, [1107](#)
  - GetFileName, [1107](#)
  - GetFileNames, [1107](#)
  - GetLossyFlagFromFile, [1107](#)
  - GetMD5DataImage, [1108](#)
  - GetMD5DataImages, [1108](#)
  - GetMD5FromBrokenFile, [1108](#)
  - GetMD5FromFile, [1108](#)
  - GetMediaStorageDataFile, [1108](#)
  - GetMediaStorageDataFiles, [1108](#)
  - GetMediaStorageFromFile, [1109](#)
  - GetNumberOfFileNames, [1109](#)
  - GetNumberOfMD5DataImages, [1109](#)
  - GetNumberOfMediaStorageDataFiles, [1109](#)
  - GetPixelSpacingDataRoot, [1109](#)
  - GetSelectedPrivateGroupOffsetFromFile, [1109](#)
  - GetSelectedTagsOffsetFromFile, [1110](#)
  - GetSourceDirectory, [1110](#)
  - GetStreamOffsetFromFile, [1110](#)
  - GetTempDirectory, [1110](#)
  - GetTempDirectoryW, [1110](#)
  - GetTempFilename, [1111](#)
  - GetTempFilenameW, [1111](#)
  - MD5DataImagesType, [1105](#)
  - MediaStorageDataFilesType, [1105](#)
  - Print, [1111](#)
  - Testing, [1106](#)
- gdcmm::Trace, [1112](#)
  - ~Trace, [1113](#)
  - DebugOff, [1113](#)
  - DebugOn, [1113](#)
  - ErrorOff, [1114](#)
  - ErrorOn, [1114](#)
  - GetDebugFlag, [1114](#)
  - GetDebugStream, [1114](#)
  - GetErrorFlag, [1114](#)
  - GetErrorStream, [1114](#)
  - GetStream, [1115](#)
  - GetWarningFlag, [1115](#)
  - GetWarningStream, [1115](#)
  - SetDebug, [1115](#)
  - SetDebugStream, [1115](#)
  - SetError, [1115](#)
  - SetErrorStream, [1116](#)
  - SetStream, [1116](#)
  - SetStreamToFile, [1116](#)
  - SetWarning, [1116](#)

- SetWarningStream, 1116
- Trace, 1113
- WarningOff, 1117
- WarningOn, 1117
- gdcmm::TransferSyntax, 1117
  - CanStoreLossy, 1121
  - GetNegociatedType, 1121
  - GetString, 1121
  - GetSwapCode, 1121
  - GetTSString, 1121
  - GetTSType, 1122
  - IsEncapsulated, 1122
  - IsEncoded, 1122
  - IsExplicit, 1122
  - IsImplicit, 1122
  - IsLossless, 1123
  - IsLossy, 1123
  - IsValid, 1123
  - NegociatedType, 1119
  - operator TSType, 1123
  - operator<<, 1123
  - TSType, 1120
  - TransferSyntax, 1120
- gdcmm::Type, 1128
  - GetTypeString, 1130
  - GetTypeType, 1130
  - operator TypeType, 1130
  - operator<<, 1130
  - Type, 1130
  - TypeType, 1129
- gdcmm::UIDGenerator, 1132
  - Generate, 1133
  - GenerateUUID, 1133
  - GetGDCMUID, 1133
  - GetRoot, 1133
  - IsValid, 1134
  - SetRoot, 1134
  - UIDGenerator, 1133
- gdcmm::UIDs, 1134
  - GetName, 1158
  - GetNumberOfTransferSyntaxStrings, 1158
  - GetString, 1159
  - GetTransferSyntaxString, 1159
  - GetTransferSyntaxStrings, 1159
  - GetUIDName, 1159
  - GetUIDString, 1159
  - operator TSType, 1159
  - SetFromUID, 1160
  - TSName, 1145
  - TSType, 1152
  - TransferSyntaxStringsType, 1145
- gdcmm::UNExplicitDataElement, 1227
  - GetLength, 1228
  - Read, 1229
- ReadPreValue, 1229
- ReadValue, 1229
- ReadWithLength, 1229
- gdcmm::UNExplicitImplicitDataElement, 1230
  - GetLength, 1231
  - Read, 1231
  - ReadPreValue, 1231
  - ReadValue, 1231
- gdcmm::UUIDGenerator, 1240
  - Generate, 1240
  - IsValid, 1241
- gdcmm::UI, 1131
  - Internal, 1131
  - operator<<, 1131
- gdcmm::Unpacker12Bits, 1232
  - Pack, 1232
  - Unpack, 1233
- gdcmm::Usage, 1233
  - GetUsageString, 1235
  - GetUsageType, 1235
  - operator UsageType, 1235
  - operator<<, 1235
  - Usage, 1235
  - UsageType, 1234
- gdcmm::UserEvent, 1236
- gdcmm::VMToLength< T >, 1260
- gdcmm::VR16ExplicitDataElement, 1268
  - GetLength, 1269
  - Read, 1269
  - ReadPreValue, 1269
  - ReadValue, 1269
  - ReadWithLength, 1270
- gdcmm::VRToEncoding< T >, 1270
- gdcmm::VRToType< T >, 1270
- gdcmm::VRVLSIZE< 0 >, 1271
  - Read, 1271
  - Write, 1271
- gdcmm::VRVLSIZE< 1 >, 1272
  - Read, 1272
  - Write, 1272
- gdcmm::VRVLSIZE< T >, 1271
- gdcmm::Validate, 1241
  - ~Validate, 1242
  - F, 1243
  - GetValidatedFile, 1242
  - SetFile, 1242
  - V, 1243
  - Validate, 1242
  - Validation, 1243
- gdcmm::Value, 1244
  - ~Value, 1245
  - Clear, 1245
  - DataElement, 1246
  - GetLength, 1245

- operator==, [1246](#)
- SetLength, [1246](#)
- SetLengthOnly, [1246](#)
- Value, [1245](#)
- gdcm::ValueIO< TDE, TSwap, TType >, [1247](#)
- gdcm::ValueIO
  - Read, [1247](#)
  - Write, [1247](#)
- gdcm::Version, [1248](#)
  - ~Version, [1249](#)
  - GetBuildVersion, [1249](#)
  - GetMajorVersion, [1250](#)
  - GetMinorVersion, [1250](#)
  - GetVersion, [1250](#)
  - operator<<, [1250](#)
  - Print, [1250](#)
  - Version, [1249](#)
- gdcm::VL, [1251](#)
  - GetLength, [1252](#)
  - GetVL16Max, [1252](#)
  - GetVL32Max, [1253](#)
  - IsOdd, [1253](#)
  - IsUndefined, [1253](#)
  - operator uint32\_t, [1253](#)
  - operator<<, [1255](#)
  - operator++, [1253](#)
  - operator+=, [1253](#)
  - Read, [1254](#)
  - Read16, [1254](#)
  - SetToUndefined, [1254](#)
  - Type, [1252](#)
  - VL, [1252](#)
  - Write, [1254](#)
  - Write16, [1254](#)
- gdcm::VM, [1255](#)
  - Compatible, [1258](#)
  - GetIndex, [1258](#)
  - GetLength, [1258](#)
  - GetNumberOfElementsFromArray, [1258](#)
  - GetVMString, [1258](#)
  - GetVMType, [1259](#)
  - GetVMTypeFromLength, [1259](#)
  - IsValid, [1259](#)
  - operator VMType, [1259](#)
  - operator<<, [1259](#)
  - VMType, [1257](#)
  - VM, [1258](#)
- gdcm::VR, [1260](#)
  - CanDisplay, [1263](#)
  - Compatible, [1264](#)
  - GetLength, [1264](#)
  - GetSize, [1264](#)
  - GetSizeof, [1264](#)
  - GetVRString, [1264](#)
  - GetVRStringFromFile, [1265](#)
  - GetVRType, [1265](#)
  - GetVRTypeFromFile, [1265](#)
  - IsASCII2, [1265](#)
  - IsASCII, [1265](#)
  - IsBinary, [1265](#)
  - IsBinary2, [1266](#)
  - IsDual, [1266](#)
  - IsSwap, [1266](#)
  - IsVRFile, [1266](#)
  - IsValid, [1266](#)
  - operator VRType, [1267](#)
  - operator<<, [1267](#)
  - Read, [1267](#)
  - VRType, [1262](#)
  - VR, [1263](#)
  - Write, [1267](#)
- gdcm::WLMFindQuery, [1390](#)
  - GetAbstractSyntaxUID, [1392](#)
  - GetTagListByLevel, [1392](#)
  - GetValidDataSet, [1392](#)
  - InitializeDataSet, [1392](#)
  - QueryFactory, [1393](#)
  - ValidateQuery, [1392](#)
  - WLMFindQuery, [1391](#)
- gdcm::Waveform, [1389](#)
  - Waveform, [1390](#)
- gdcm::Writer, [1393](#)
  - ~Writer, [1396](#)
  - CheckFileMetaInformationOff, [1396](#)
  - CheckFileMetaInformationOn, [1396](#)
  - GetCheckFileMetaInformation, [1397](#)
  - GetFile, [1397](#)
  - GetStreamPtr, [1397](#)
  - Ofstream, [1399](#)
  - SetCheckFileMetaInformation, [1397](#)
  - SetFile, [1397](#)
  - SetFileName, [1398](#)
  - SetStream, [1398](#)
  - SetWriteDataSetOnly, [1398](#)
  - Stream, [1399](#)
  - StreamImageWriter, [1399](#)
  - Write, [1398](#)
  - Writer, [1396](#)
- gdcm::XMLDictReader, [1400](#)
  - ~XMLDictReader, [1401](#)
  - CharacterDataHandler, [1401](#)
  - EndElement, [1401](#)
  - GetDict, [1402](#)
  - HandleDescription, [1402](#)
  - HandleEntry, [1402](#)
  - StartElement, [1402](#)
  - XMLDictReader, [1401](#)
- gdcm::XMLPrinter, [1403](#)

- ~XMLPrinter, [1404](#)
- F, [1406](#)
- GetPrintStyle, [1404](#)
- HandleBulkData, [1404](#)
- Print, [1405](#)
- PrintDataElement, [1405](#)
- PrintDataSet, [1405](#)
- PrintSQ, [1405](#)
- PrintStyle, [1406](#)
- PrintStyles, [1404](#)
- SetFile, [1405](#)
- SetStyle, [1406](#)
- XMLPrinter, [1404](#)
- gdcmm::XMLPrivateDictReader, [1407](#)
- ~XMLPrivateDictReader, [1408](#)
- CharacterDataHandler, [1408](#)
- EndElement, [1408](#)
- GetPrivateDict, [1409](#)
- HandleDescription, [1409](#)
- HandleEntry, [1409](#)
- StartElement, [1409](#)
- XMLPrivateDictReader, [1408](#)
- gdcmm::ignore\_char, [515](#)
- ignore\_char, [515](#)
- m\_char, [516](#)
- gdcmm::network, [77](#)
- cMaxEventID, [83](#)
- cMaxStateID, [83](#)
- EEventID, [81](#)
- EStateID, [82](#)
- GetStateIndex, [83](#)
- gdcmm::network::AAAbortPDU, [87](#)
- AAAbortPDU, [88](#)
- IsLastFragment, [88](#)
- Print, [88](#)
- Read, [88](#)
- SetReason, [89](#)
- SetSource, [89](#)
- Size, [89](#)
- Write, [89](#)
- gdcmm::network::AAssociateACPD, [90](#)
- AAssociateACPD, [91](#)
- AAssociateRQPD, [94](#)
- AddPresentationContextAC, [92](#)
- GetNumberOfPresentationContextAC, [92](#)
- GetPresentationContextAC, [92](#)
- GetUserInformation, [92](#)
- InitFromRQ, [92](#)
- IsLastFragment, [92](#)
- Print, [93](#)
- Read, [93](#)
- SetCalledAETitle, [93](#)
- SetCallingAETitle, [93](#)
- Size, [93](#)
- SizeType, [91](#)
- Write, [94](#)
- gdcmm::network::AAssociateRJPD, [94](#)
- AAssociateRJPD, [95](#)
- IsLastFragment, [95](#)
- Print, [96](#)
- Read, [96](#)
- Size, [96](#)
- Write, [96](#)
- gdcmm::network::AAssociateRQPD, [97](#)
- AAssociateACPD, [103](#)
- AAssociateRQPD, [99](#)
- AddPresentationContext, [99](#)
- GetCalledAETitle, [99](#)
- GetCallingAETitle, [99](#)
- GetNumberOfPresentationContext, [100](#)
- GetPresentationContext, [100](#)
- GetPresentationContextByAbstractSyntax, [100](#)
- GetPresentationContextByID, [100](#)
- GetPresentationContexts, [100](#)
- GetReserved43\_74, [100](#)
- GetUserInformation, [101](#)
- IsAETitleValid, [101](#)
- IsLastFragment, [101](#)
- PresentationContextArrayType, [98](#)
- Print, [101](#)
- Read, [101](#)
- SetCalledAETitle, [102](#)
- SetCallingAETitle, [102](#)
- SetUserInformation, [102](#)
- Size, [102](#)
- SizeType, [99](#)
- Write, [102](#)
- gdcmm::network::ARTIMTimer, [129](#)
- ARTIMTimer, [129](#)
- GetElapsedTime, [129](#)
- GetHasExpired, [130](#)
- GetTimeout, [130](#)
- SetTimeout, [130](#)
- Start, [130](#)
- Stop, [130](#)
- gdcmm::network::AReleaseRPPD, [124](#)
- AReleaseRPPD, [125](#)
- IsLastFragment, [125](#)
- Print, [125](#)
- Read, [125](#)
- Size, [126](#)
- Write, [126](#)
- gdcmm::network::AReleaseRQPD, [126](#)
- AReleaseRQPD, [127](#)
- IsLastFragment, [127](#)
- Print, [128](#)
- Read, [128](#)
- Size, [128](#)

Write, 128  
 gdcmm::network::AbstractSyntax, 104  
   AbstractSyntax, 105  
   GetAsDataElement, 105  
   GetName, 105  
   operator==, 105  
   Print, 105  
   Read, 106  
   SetName, 106  
   SetNameFromUID, 106  
   Size, 106  
   Write, 106  
 gdcmm::network::ApplicationContext, 119  
   ApplicationContext, 120  
   GetName, 120  
   Print, 120  
   Read, 120  
   SetName, 120  
   Size, 120  
   Write, 121  
 gdcmm::network::AsynchronousOperationsWindowSub, 132  
   AsynchronousOperationsWindowSub, 132  
   Print, 133  
   Read, 133  
   Size, 133  
   Write, 133  
 gdcmm::network::BaseCompositeMessage, 169  
   ~BaseCompositeMessage, 170  
   ConstructPDV, 170  
 gdcmm::network::BaseNormalizedMessage, 171  
   ~BaseNormalizedMessage, 172  
   ConstructPDV, 172  
 gdcmm::network::BasePDU, 173  
   ~BasePDU, 174  
   IsLastFragment, 174  
   Print, 175  
   Read, 175  
   Size, 175  
   Write, 175  
 gdcmm::network::CEchoRSP, 236  
   ConstructPDVByDataSet, 237  
 gdcmm::network::CEchoRQ, 234  
   AffectedSOPClassUID, 236  
   ConstructPDV, 235  
   MessageID, 236  
 gdcmm::network::CFind, 237  
 gdcmm::network::CFindCancelRQ, 238  
   ConstructPDVByDataSet, 239  
 gdcmm::network::CFindRSP, 241  
   ConstructPDVByDataSet, 242  
 gdcmm::network::CFindRQ, 239  
   ConstructPDV, 240  
 gdcmm::network::CMoveCancelRq, 242  
   ConstructPDVByDataSet, 243  
 gdcmm::network::CMoveRSP, 245  
   ConstructPDVByDataSet, 246  
 gdcmm::network::CMoveRQ, 243  
   ConstructPDV, 244  
 gdcmm::network::CStoreRSP, 298  
   ConstructPDV, 299  
 gdcmm::network::CStoreRQ, 297  
   ConstructPDV, 298  
 gdcmm::network::CompositeMessageFactory, 259  
   ConstructCEchoRQ, 260  
   ConstructCFindRQ, 260  
   ConstructCMoveRQ, 260  
   ConstructCStoreRSP, 261  
   ConstructCStoreRQ, 261  
 gdcmm::network::DIMSE, 376  
   CommandTypes, 377  
 gdcmm::network::ImplementationClassUIDSub, 578  
   ImplementationClassUIDSub, 579  
   Print, 579  
   Read, 579  
   Size, 579  
   Write, 580  
 gdcmm::network::ImplementationUIDSub, 580  
   ImplementationUIDSub, 580  
   Write, 581  
 gdcmm::network::ImplementationVersionNameSub, 581  
   ImplementationVersionNameSub, 581  
   Print, 582  
   Read, 582  
   Size, 582  
   Write, 582  
 gdcmm::network::MaximumLengthSub, 662  
   GetMaximumLength, 663  
   MaximumLengthSub, 662  
   Print, 663  
   Read, 663  
   SetMaximumLength, 663  
   Size, 663  
   Write, 664  
 gdcmm::network::NActionRSP, 713  
   ConstructPDVByDataSet, 714  
 gdcmm::network::NActionRQ, 712  
   ConstructPDV, 713  
 gdcmm::network::NCreateRSP, 716  
   ConstructPDVByDataSet, 717  
 gdcmm::network::NCreateRQ, 715  
   ConstructPDV, 716  
 gdcmm::network::NDeleteRSP, 719  
   ConstructPDVByDataSet, 720  
 gdcmm::network::NDeleteRQ, 718  
   ConstructPDV, 719  
 gdcmm::network::NEventReportRSP, 725  
   ConstructPDVByDataSet, 726

- gdcmm::network::NEventReportRQ, 724
  - ConstructPDV, 725
- gdcmm::network::NGetRSP, 728
  - ConstructPDVByDataSet, 729
- gdcmm::network::NGetRQ, 727
  - ConstructPDV, 728
- gdcmm::network::NSetRSP, 737
  - ConstructPDVByDataSet, 738
- gdcmm::network::NSetRQ, 735
  - ConstructPDV, 736
- gdcmm::network::NormalizedMessageFactory, 731
  - ConstructNAction, 731
  - ConstructNCreate, 731
  - ConstructNDelete, 731
  - ConstructNEventReport, 732
  - ConstructNGet, 732
  - ConstructNSet, 732
- gdcmm::network::PDUFactory, 785
  - ConstructAbortPDU, 786
  - ConstructPDU, 786
  - ConstructReleasePDU, 786
  - CreateCEchoPDU, 786
  - CreateCFindPDU, 787
  - CreateCMovePDU, 787
  - CreateCStoreRQPDU, 787
  - CreateCStoreRSPPDU, 787
  - CreateNActionPDU, 787
  - CreateNCreatePDU, 787
  - CreateNDeletePDU, 788
  - CreateNEventReportPDU, 788
  - CreateNGetPDU, 788
  - CreateNSetPDU, 788
  - DetermineEventByPDU, 788
  - GetPDVs, 788
- gdcmm::network::PDataTFPDU, 773
  - AddPresentationDataValue, 774
  - GetNumberOfPresentationDataValues, 774
  - GetPresentationDataValue, 775
  - IsLastFragment, 775
  - PDataTFPDU, 774
  - Print, 775
  - Read, 775
  - ReadInto, 775
  - Size, 775
  - SizeType, 774
  - Write, 776
- gdcmm::network::PresentationContextAC, 837
  - GetPresentationContextID, 838
  - GetReason, 838
  - GetTransferSyntax, 838
  - PresentationContextAC, 837
  - Print, 838
  - Read, 838
  - SetPresentationContextID, 838
  - SetReason, 839
  - SetTransferSyntax, 839
  - Size, 839
  - Write, 839
- gdcmm::network::PresentationContextRQ, 843
  - AddTransferSyntax, 845
  - GetAbstractSyntax, 845
  - GetNumberOfTransferSyntaxes, 845
  - GetPresentationContextID, 845
  - GetTransferSyntax, 845
  - GetTransferSyntaxes, 846
  - operator==, 846
  - PresentationContextRQ, 844
  - Print, 846
  - Read, 846
  - SetAbstractSyntax, 846
  - SetPresentationContextID, 846
  - Size, 847
  - SizeType, 844
  - Write, 847
- gdcmm::network::PresentationDataValue, 847
  - ConcatenatePDVBlobs, 848
  - ConcatenatePDVBlobsAsExplicit, 848
  - GetBlob, 848
  - GetIsCommand, 849
  - GetIsLastFragment, 849
  - GetMessageHeader, 849
  - GetPresentationContextID, 849
  - PresentationDataValue, 848
  - Print, 849
  - Read, 849
  - ReadInto, 849
  - SetBlob, 850
  - SetCommand, 850
  - SetDataSet, 850
  - SetLastFragment, 850
  - SetMessageHeader, 850
  - SetPresentationContextID, 851
  - Size, 851
  - Write, 851
- gdcmm::network::RoleSelectionSub, 912
  - Print, 913
  - Read, 913
  - RoleSelectionSub, 912
  - SetTuple, 913
  - Size, 913
  - Write, 913
- gdcmm::network::SOPClassExtendedNegotiationSub, 994
  - Print, 995
  - Read, 995
  - SOPClassExtendedNegotiationSub, 995
  - SetTuple, 995
  - Size, 995
  - Write, 995



- gdcmm::network::ServiceClassApplicationInformation, 969
  - Print, 969
  - Read, 969
  - ServiceClassApplicationInformation, 969
  - SetTuple, 969
  - Size, 970
  - Write, 970
- gdcmm::network::TableRow, 1089
  - ~TableRow, 1089
  - TableRow, 1089
  - transitions, 1090
- gdcmm::network::TransferSyntaxSub, 1124
  - GetName, 1124
  - operator==, 1125
  - Print, 1125
  - Read, 1125
  - SetName, 1125
  - SetNameFromUID, 1125
  - Size, 1125
  - TransferSyntaxSub, 1124
  - Write, 1126
- gdcmm::network::Transition, 1126
  - ~Transition, 1127
  - mAction, 1128
  - mEnd, 1128
  - MakeNew, 1127
  - Transition, 1127
- gdcmm::network::ULAction, 1160
  - ~ULAction, 1162
  - PerformAction, 1163
  - ULAction, 1162
- gdcmm::network::ULActionAA1, 1163
  - PerformAction, 1164
- gdcmm::network::ULActionAA2, 1165
  - PerformAction, 1165
- gdcmm::network::ULActionAA3, 1166
  - PerformAction, 1167
- gdcmm::network::ULActionAA4, 1167
  - PerformAction, 1168
- gdcmm::network::ULActionAA5, 1169
  - PerformAction, 1169
- gdcmm::network::ULActionAA6, 1170
  - PerformAction, 1171
- gdcmm::network::ULActionAA7, 1171
  - PerformAction, 1172
- gdcmm::network::ULActionAA8, 1173
  - PerformAction, 1173
- gdcmm::network::ULActionAE1, 1174
  - PerformAction, 1175
- gdcmm::network::ULActionAE2, 1175
  - PerformAction, 1176
- gdcmm::network::ULActionAE3, 1177
  - PerformAction, 1177
- gdcmm::network::ULActionAE4, 1178
  - PerformAction, 1179
- gdcmm::network::ULActionAE5, 1179
  - PerformAction, 1180
- gdcmm::network::ULActionAE6, 1181
  - PerformAction, 1181
- gdcmm::network::ULActionAE7, 1182
  - PerformAction, 1183
- gdcmm::network::ULActionAE8, 1183
  - PerformAction, 1184
- gdcmm::network::ULActionAR1, 1185
  - PerformAction, 1185
- gdcmm::network::ULActionAR10, 1186
  - PerformAction, 1187
- gdcmm::network::ULActionAR2, 1187
  - PerformAction, 1188
- gdcmm::network::ULActionAR3, 1189
  - PerformAction, 1189
- gdcmm::network::ULActionAR4, 1190
  - PerformAction, 1191
- gdcmm::network::ULActionAR5, 1191
  - PerformAction, 1192
- gdcmm::network::ULActionAR6, 1193
  - PerformAction, 1193
- gdcmm::network::ULActionAR7, 1194
  - PerformAction, 1195
- gdcmm::network::ULActionAR8, 1195
  - PerformAction, 1196
- gdcmm::network::ULActionAR9, 1197
  - PerformAction, 1197
- gdcmm::network::ULActionDT1, 1198
  - PerformAction, 1199
- gdcmm::network::ULActionDT2, 1199
  - PerformAction, 1200
- gdcmm::network::ULBasicCallback, 1201
  - ~ULBasicCallback, 1202
  - GetDataSets, 1202
  - GetResponses, 1202
  - HandleDataSet, 1202
  - HandleResponse, 1203
  - ULBasicCallback, 1202
- gdcmm::network::ULConnection, 1203
  - ~ULConnection, 1204
  - AddAcceptedPresentationContext, 1204
  - FindContext, 1205
  - GetAcceptedPresentationContexts, 1205
  - GetConnectionInfo, 1205
  - GetMaxPDUSize, 1205
  - GetPresentationContextACByID, 1205
  - GetPresentationContextIDFromPresentationContext, 1205
  - GetPresentationContextRQByID, 1206
  - GetPresentationContexts, 1206
  - GetProtocol, 1206
  - GetState, 1206



- GetTimer, [1206](#)
- InitializeConnection, [1206](#)
- InitializeIncomingConnection, [1207](#)
- SetMaxPDUSize, [1207](#)
- SetPresentationContexts, [1207](#)
- SetState, [1207](#)
- StopProtocol, [1207](#)
- ULActionAE6, [1208](#)
- ULConnection, [1204](#)
- ULConnectionManager, [1208](#)
- gdcmm::network::ULConnectionCallback, [1208](#)
  - ~ULConnectionCallback, [1209](#)
  - DataSetHandled, [1210](#)
  - DataSetHandles, [1210](#)
  - HandleDataSet, [1210](#)
  - HandleResponse, [1210](#)
  - mImplicit, [1211](#)
  - ResetHandledDataSet, [1210](#)
  - SetImplicitFlag, [1210](#)
  - ULConnectionCallback, [1209](#)
- gdcmm::network::ULConnectionInfo, [1211](#)
  - GetCalledAETitle, [1212](#)
  - GetCalledComputerName, [1212](#)
  - GetCalledIPAddress, [1212](#)
  - GetCalledIPPort, [1212](#)
  - GetCallingAETitle, [1212](#)
  - GetMaxPDULength, [1212](#)
  - Initialize, [1212](#)
  - SetMaxPDULength, [1213](#)
  - ULConnectionInfo, [1211](#)
- gdcmm::network::ULConnectionManager, [1213](#)
  - ~ULConnectionManager, [1215](#)
  - BreakConnection, [1216](#)
  - BreakConnectionNow, [1216](#)
  - EstablishConnection, [1216](#)
  - EstablishConnectionMove, [1216](#)
  - mConnection, [1220](#)
  - mSecondaryConnection, [1220](#)
  - mTransitions, [1220](#)
  - RunEventLoop, [1216](#)
  - RunMoveEventLoop, [1217](#)
  - SendEcho, [1217](#)
  - SendFind, [1217](#)
  - SendMove, [1217](#)
  - SendNAction, [1218](#)
  - SendNCreate, [1218](#)
  - SendNDelete, [1218](#)
  - SendNEventReport, [1219](#)
  - SendNGet, [1219](#)
  - SendNSet, [1219](#)
  - SendStore, [1220](#)
  - ULConnectionManager, [1215](#)
- gdcmm::network::ULEvent, [1221](#)
  - ~ULEvent, [1222](#)
- GetDataSetPos, [1222](#)
- GetEvent, [1222](#)
- GetIStream, [1222](#)
- GetPDUs, [1222](#)
- SetEvent, [1223](#)
- SetPDU, [1223](#)
- ULEvent, [1221](#), [1222](#)
- gdcmm::network::ULTransitionTable, [1223](#)
  - HandleEvent, [1224](#)
  - PrintTable, [1224](#)
  - ULTransitionTable, [1224](#)
- gdcmm::network::ULWritingCallback, [1225](#)
  - ~ULWritingCallback, [1226](#)
  - HandleDataSet, [1226](#)
  - HandleResponse, [1226](#)
  - SetDirectory, [1226](#)
  - ULWritingCallback, [1226](#)
- gdcmm::network::UserInformation, [1237](#)
  - ~UserInformation, [1238](#)
  - AddRoleSelectionSub, [1238](#)
  - AddSOPClassExtendedNegociationSub, [1238](#)
  - GetMaximumLengthSub, [1239](#)
  - operator=, [1239](#)
  - Print, [1239](#)
  - Read, [1239](#)
  - Size, [1239](#)
  - UserInformation, [1238](#)
  - Write, [1240](#)
- gdcmm::static\_assert\_test< x >, [1010](#)
- gdcmm::terminal, [84](#)
  - Attribute, [85](#)
  - Color, [85](#)
  - Mode, [85](#)
  - setAttribute, [86](#)
  - setBackground, [86](#)
  - setfgcolor, [86](#)
  - setmode, [86](#)
- gdcmmAAabortPDU.h, [1411](#)
- gdcmmAAAssociateACPDU.h, [1412](#)
- gdcmmAAAssociateRJPDU.h, [1412](#)
- gdcmmAAAssociateRQPDU.h, [1413](#)
- gdcmmARTIMTimer.h, [1420](#)
- gdcmmAReleaseRPPDU.h, [1418](#)
- gdcmmAReleaseRQPDU.h, [1419](#)
- gdcmmASN1.h, [1421](#)
- gdcmmAbstractSyntax.h, [1414](#)
- gdcmmAnonymizeEvent.h, [1415](#)
- gdcmmAnonymizer.h, [1416](#)
- gdcmmApplicationContext.h, [1417](#)
- gdcmmApplicationEntity.h, [1418](#)
- gdcmmAssertAlwaysMacro
  - gdcmmTrace.h, [1647](#)
- gdcmmAssertMacro
  - gdcmmTrace.h, [1647](#)

gdcmAsynchronousOperationsWindowSub.h, 1422  
gdcmAttribute.h, 1422  
gdcmAudioCodec.h, 1424  
gdcmBase64.h, 1424  
gdcmBaseCompositeMessage.h, 1425  
gdcmBaseNormalizedMessage.h, 1426  
gdcmBasePDU.h, 1427  
gdcmBaseQuery.h, 1428  
gdcmBaseRootQuery.h, 1429  
gdcmBasicOffsetTable.h, 1430  
gdcmBitmap.h, 1431  
gdcmBitmapToBitmapFilter.h, 1432  
gdcmBoxRegion.h, 1433  
gdcmByteBuffer.h, 1433  
gdcmByteSwap.h, 1435  
gdcmByteSwapFilter.h, 1435  
gdcmByteValue.h, 1436  
gdcmCAPICryptoFactory.h, 1437  
gdcmCAPICryptographicMessageSyntax.h, 1438  
gdcmCEchoMessages.h, 1438  
gdcmCFindMessages.h, 1439  
gdcmCMoveMessages.h, 1440  
gdcmCP246ExplicitDataElement.h, 1448  
gdcmCSAElement.h, 1451  
gdcmCSAHeader.h, 1452  
gdcmCSAHeaderDict.h, 1453  
gdcmCSAHeaderDictEntry.h, 1454  
gdcmCStoreMessages.h, 1455  
gdcmCodeString.h, 1444  
gdcmCodec.h, 1441  
gdcmCoder.h, 1442  
gdcmCommand.h, 1445  
gdcmCommandDataSet.h, 1446  
gdcmCompositeMessageFactory.h, 1447  
gdcmCompositeNetworkFunctions.h, 1447  
gdcmConstCharWrapper.h, 1448  
gdcmCryptoFactory.h, 1449  
gdcmCryptographicMessageSyntax.h, 1450  
gdcmCurve.h, 1456  
gdcmDICOMDIR.h, 1466  
gdcmDICOMDIRGenerator.h, 1467  
gdcmDIMSE.h, 1473  
gdcmDataElement.h, 1457  
gdcmDataEvent.h, 1459  
gdcmDataSet.h, 1460  
gdcmDataSetEvent.h, 1461  
gdcmDataSetHelper.h, 1461  
gdcmDebugMacro  
    gdcmTrace.h, 1648  
gdcmDecoder.h, 1462  
gdcmDefinedTerms.h, 1464  
gdcmDeflateStream.h, 1464  
gdcmDefs.h, 1465  
gdcmDeltaEncodingCodec.h, 1466  
gdcmDict.h, 1468  
gdcmDictConverter.h, 1469  
gdcmDictEntry.h, 1470  
gdcmDictPrinter.h, 1471  
gdcmDicts.h, 1472  
gdcmDirectionCosines.h, 1473  
gdcmDirectory.h, 1474  
gdcmDirectoryHelper.h, 1475  
gdcmDummyValueGenerator.h, 1476  
gdcmDumper.h, 1476  
gdcmElement.h, 1477  
    VRDS16ILLEGAL, 1479  
gdcmEmptyMaskGenerator.h, 1479  
gdcmEncapsulatedDocument.h, 1480  
gdcmEnumeratedValues.h, 1481  
gdcmErrorMacro  
    gdcmTrace.h, 1648  
gdcmEvent.h, 1481  
    gdcmEventMacro, 1482  
gdcmEventMacro  
    gdcmEvent.h, 1482  
gdcmException.h, 1483  
gdcmExplicitDataElement.h, 1484  
gdcmExplicitImplicitDataElement.h, 1485  
gdcmFiducials.h, 1485  
gdcmFile.h, 1486  
gdcmFileAnonymizer.h, 1487  
gdcmFileChangeTransferSyntax.h, 1488  
gdcmFileDecompressLookupTable.h, 1488  
gdcmFileDerivation.h, 1489  
gdcmFileExplicitFilter.h, 1490  
gdcmFileMetaInformation.h, 1491  
gdcmFileNameEvent.h, 1492  
gdcmFileSet.h, 1494  
gdcmFileStreamer.h, 1495  
gdcmFilename.h, 1492  
gdcmFilenameGenerator.h, 1493  
gdcmFindPatientRootQuery.h, 1496  
gdcmFindStudyRootQuery.h, 1497  
gdcmFragment.h, 1497  
gdcmGlobal.h, 1499  
gdcmGroupDict.h, 1500  
gdcmIOD.h, 1517  
gdcmIODEntry.h, 1519  
gdcmIODs.h, 1521  
gdcmIPPSorter.h, 1522  
gdcmIconImage.h, 1500  
gdcmIconImageFilter.h, 1501  
gdcmIconImageGenerator.h, 1502  
gdcmImage.h, 1503  
gdcmImageApplyLookupTable.h, 1504  
gdcmImageChangePhotometricInterpretation.h, 1505  
gdcmImageChangePlanarConfiguration.h, 1506  
gdcmImageChangeTransferSyntax.h, 1506

gdcmImageCodec.h, 1507  
gdcmImageConverter.h, 1508  
gdcmImageFragmentSplitter.h, 1509  
gdcmImageHelper.h, 1510  
gdcmImageReader.h, 1511  
gdcmImageRegionReader.h, 1512  
gdcmImageToImageFilter.h, 1512  
gdcmImageWriter.h, 1513  
gdcmImplementationClassUIDSub.h, 1514  
gdcmImplementationUIDSub.h, 1515  
gdcmImplementationVersionNameSub.h, 1515  
gdcmImplicitDataElement.h, 1517  
gdcmItem.h, 1523  
gdcmJPEG12Codec.h, 1524  
gdcmJPEG16Codec.h, 1525  
gdcmJPEG2000Codec.h, 1525  
gdcmJPEG8Codec.h, 1526  
gdcmJPEGCodec.h, 1527  
gdcmJPEGLSCodec.h, 1528  
gdcmJSON.h, 1528  
gdcmKAKADUCodec.h, 1529  
gdcmLO.h, 1531  
gdcmLegacyMacro.h, 1530  
    GDCM\_LEGACY\_BODY, 1531  
    GDCM\_LEGACY\_REPLACED\_BODY, 1531  
    GDCM\_LEGACY, 1530  
gdcmLookupTable.h, 1532  
gdcmMD5.h, 1540  
gdcmMacro.h, 1533  
gdcmMacroEntry.h, 1535  
    GDCMMACROENTRY\_H, 1536  
gdcmMacros.h, 1537  
gdcmMaxLengthSub.h, 1538  
gdcmMediaStorage.h, 1540  
gdcmMeshPrimitive.h, 1542  
gdcmModalityPerformedProcedureStepCreateQuery.h, 1543  
gdcmModalityPerformedProcedureStepSetQuery.h, 1544  
gdcmModule.h, 1544  
gdcmModuleEntry.h, 1546  
gdcmModules.h, 1548  
gdcmMovePatientRootQuery.h, 1549  
gdcmMoveStudyRootQuery.h, 1550  
gdcmMrProtocol.h, 1550  
gdcmNActionMessages.h, 1552  
gdcmNCreateMessages.h, 1553  
gdcmNDeleteMessages.h, 1553  
gdcmNEventReportMessages.h, 1557  
gdcmNGetMessages.h, 1558  
gdcmNSetMessages.h, 1560  
gdcmNestedModuleEntries.h, 1554  
gdcmNetworkEvents.h, 1555  
gdcmNetworkStateID.h, 1556  
gdcmNormalizedMessageFactory.h, 1558  
gdcmNormalizedNetworkFunctions.h, 1559  
gdcmObject.h, 1560  
gdcmOpenSSLCryptoFactory.h, 1562  
gdcmOpenSSLCryptographicMessageSyntax.h, 1562  
gdcmOpenSSL7CryptoFactory.h, 1564  
gdcmOpenSSL7CryptographicMessageSyntax.h, 1564  
gdcmOrientation.h, 1566  
gdcmOverlay.h, 1566  
gdcmPDSElement.h, 1571  
gdcmPDBHeader.h, 1572  
gdcmPDFCodec.h, 1573  
gdcmPDUFactory.h, 1574  
gdcmPDataTFPDU.h, 1570  
gdcmPGXCodec.h, 1575  
gdcmPNMCodec.h, 1582  
gdcmPVRGCodec.h, 1592  
gdcmParseException.h, 1567  
gdcmParser.h, 1569  
gdcmPatient.h, 1569  
gdcmPersonName.h, 1574  
gdcmPhotometricInterpretation.h, 1576  
gdcmPixelFormat.h, 1577  
gdcmPixmap.h, 1578  
gdcmPixmapReader.h, 1579  
gdcmPixmapToPixmapFilter.h, 1580  
gdcmPixmapWriter.h, 1580  
gdcmPreamble.h, 1582  
gdcmPresentationContext.h, 1584  
gdcmPresentationContextAC.h, 1585  
gdcmPresentationContextGenerator.h, 1586  
gdcmPresentationContextRQ.h, 1586  
gdcmPresentationDataValue.h, 1587  
gdcmPrinter.h, 1588  
gdcmPrivateTag.h, 1590  
gdcmProgressEvent.h, 1591  
gdcmPythonFilter.h, 1592  
gdcmQueryBase.h, 1593  
gdcmQueryFactory.h, 1594  
gdcmQueryImage.h, 1595  
gdcmQueryPatient.h, 1596  
gdcmQuerySeries.h, 1597  
gdcmQueryStudy.h, 1598  
gdcmRAWCodec.h, 1599  
gdcmRLECodec.h, 1603  
gdcmReader.h, 1599  
gdcmRegion.h, 1601  
gdcmRescaler.h, 1602  
gdcmRoleSelectionSub.h, 1603  
gdcmSHA1.h, 1615  
gdcmSOPClassExtendedNegotiationSub.h, 1618  
gdcmSOPClassUIDToIOD.h, 1619  
gdcmScanner.h, 1604  
gdcmSegment.h, 1605  
gdcmSegmentHelper.h, 1607

gdcSegmentReader.h, 1608  
 gdcSegmentWriter.h, 1609  
 gdcSegmentedPaletteColorLookupTable.h, 1606  
 gdcSequenceOfFragments.h, 1610  
 gdcSequenceOfItems.h, 1611  
 gdcSerieHelper.h, 1611  
 gdcSeries.h, 1613  
 gdcServiceClassApplicationInformation.h, 1614  
 gdcServiceClassUser.h, 1615  
 gdcSimpleSubjectWatcher.h, 1616  
 gdcSmartPointer.h, 1617  
 gdcSorter.h, 1620  
 gdcSpacing.h, 1622  
 gdcSpectroscopy.h, 1622  
 gdcSplitMosaicFilter.h, 1623  
 gdcStaticAssert.h, 1623  
     GDCM\_DO\_JOIN2, 1624  
     GDCM\_DO\_JOIN, 1624  
     GDCM\_JOIN, 1624  
     GDCM\_STATIC\_ASSERT, 1624  
 gdcStreamImageReader.h, 1625  
 gdcStreamImageWriter.h, 1626  
 gdcStrictScanner.h, 1626  
 gdcString.h, 1627  
 gdcStringFilter.h, 1629  
 gdcStudy.h, 1629  
 gdcSubject.h, 1630  
 gdcSurface.h, 1631  
 gdcSurfaceHelper.h, 1632  
 gdcSurfaceReader.h, 1633  
 gdcSurfaceWriter.h, 1634  
 gdcSwapCode.h, 1634  
 gdcSwapper.h, 1635  
 gdcSystem.h, 1636  
 gdcTable.h, 1637  
 gdcTableEntry.h, 1638  
 gdcTableReader.h, 1640  
 gdcTag.h, 1641  
 gdcTagPath.h, 1642  
 gdcTagToVR.h, 1642  
 gdcTerminal.h, 1643  
 gdcTestDriver.h, 1644  
 gdcTesting.h, 1645  
 gdcTrace.h, 1646  
     GDCM\_FUNCTION, 1647  
     gdcAssertAlwaysMacro, 1647  
     gdcAssertMacro, 1647  
     gdcDebugMacro, 1648  
     gdcErrorMacro, 1648  
     gdcWarningMacro, 1649  
 gdcTransferSyntax.h, 1650  
 gdcTransferSyntaxSub.h, 1651  
 gdcType.h, 1652  
 gdcTypes.h, 1653  
 gdcUIDGenerator.h, 1653  
 gdcUIDs.h, 1654  
 gdcULAction.h, 1655  
 gdcULActionAA.h, 1656  
 gdcULActionAE.h, 1656  
 gdcULActionAR.h, 1657  
 gdcULActionDT.h, 1658  
 gdcULBasicCallback.h, 1658  
 gdcULConnection.h, 1659  
 gdcULConnectionCallback.h, 1660  
 gdcULConnectionInfo.h, 1661  
 gdcULConnectionManager.h, 1662  
 gdcULEvent.h, 1663  
 gdcULTransitionTable.h, 1664  
 gdcULWritingCallback.h, 1665  
 gdcUNExplicitDataElement.h, 1666  
 gdcUNExplicitImplicitDataElement.h, 1666  
 gdcUUIDGenerator.h, 1670  
 gdcUnpacker12Bits.h, 1667  
 gdcUsage.h, 1667  
 gdcUserInformation.h, 1669  
 gdcVL.h, 1674  
 gdcVM.h, 1675  
     TYPETOLENGTH, 1676  
 gdcVR.h, 1676  
     TYPETOENCODING, 1678  
     VRTypeTemplateCase, 1678  
 gdcVR16ExplicitDataElement.h, 1678  
 gdcValidate.h, 1671  
 gdcValue.h, 1671  
 gdcValueIO.h, 1672  
 gdcVersion.h, 1673  
 gdcWLMFindQuery.h, 1680  
 gdcWarningMacro  
     gdcTrace.h, 1649  
 gdcWaveform.h, 1679  
 gdcWin32.h, 1679  
     GDCM\_EXPORT, 1680  
 gdcWriter.h, 1681  
 gdcXMLDictReader.h, 1682  
 gdcXMLPrinter.h, 1682  
 gdcXMLPrivateDictReader.h, 1683  
 Generate  
     gdc::DICOMDIRGenerator, 355  
     gdc::DummyValueGenerator, 389  
     gdc::FilenameGenerator, 480  
     gdc::IconImageGenerator, 513  
     gdc::UIDGenerator, 1133  
     gdc::UUIDGenerator, 1240  
 GenerateFromFilenames  
     gdc::PresentationContextGenerator, 841  
 GenerateFromUID  
     gdc::PresentationContextGenerator, 842  
 GenerateUUID

- gdcmm::UIDGenerator, [1133](#)
- Get
  - gdcmm::ByteBuffer, [216](#)
- GetAETitle
  - gdcmm::ServiceClassUser, [973](#)
- GetALGOType
  - gdcmm::Segment, [928](#)
- GetALGOTypeString
  - gdcmm::Segment, [928](#)
- GetAbbreviation
  - gdcmm::GroupDict, [507](#)
- GetAbstractSyntax
  - gdcmm::PresentationContext, [835](#)
  - gdcmm::network::PresentationContextRQ, [845](#)
- GetAbstractSyntaxUID
  - gdcmm::BaseQuery, [178](#)
  - gdcmm::FindPatientRootQuery, [494](#)
  - gdcmm::FindStudyRootQuery, [497](#)
  - gdcmm::ModalityPerformedProcedureStepCreateQuery, [688](#)
  - gdcmm::ModalityPerformedProcedureStepSetQuery, [691](#)
  - gdcmm::MovePatientRootQuery, [705](#)
  - gdcmm::MoveStudyRootQuery, [708](#)
  - gdcmm::WLMFindQuery, [1392](#)
- GetAcceptedPresentationContexts
  - gdcmm::network::ULConnection, [1205](#)
- GetAlgorithmFamily
  - gdcmm::Surface, [1049](#)
- GetAlgorithmName
  - gdcmm::Surface, [1049](#)
- GetAlgorithmVersion
  - gdcmm::Surface, [1049](#)
- GetAllFilenamesFromTagToValue
  - gdcmm::Scanner, [920](#)
  - gdcmm::StrictScanner, [1028](#)
- GetAllRequiredTags
  - gdcmm::QueryBase, [872](#)
- GetAllTags
  - gdcmm::QueryBase, [872](#)
- GetAnatomicRegion
  - gdcmm::Segment, [929](#)
- GetAnatomicRegionModifiers
  - gdcmm::Segment, [929](#)
- GetAsDataElement
  - gdcmm::Attribute, [137](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [154](#)
  - gdcmm::Element, [393](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [400](#)
  - gdcmm::PrivateTag, [861](#)
  - gdcmm::network::AbstractSyntax, [105](#)
- GetAsPoints
  - gdcmm::Curve, [302](#)
- GetAsString
  - gdcmm::CodeString, [253](#)
- GetAxisOfRotation
  - gdcmm::Surface, [1050](#)
- GetBasicApplicationLevelConfidentialityProfileAttributes
  - gdcmm::Anonymizer, [114](#)
- GetBitPosition
  - gdcmm::Overlay, [759](#)
- GetBitSample
  - gdcmm::LookupTable, [651](#)
- GetBitsAllocated
  - gdcmm::Overlay, [760](#)
  - gdcmm::PixelFormat, [801](#)
- GetBitsStored
  - gdcmm::PixelFormat, [802](#)
- GetBlob
  - gdcmm::network::PresentationDataValue, [848](#)
- GetBuffer
  - gdcmm::Bitmap, [196](#)
  - gdcmm::ByteValue, [224](#)
  - gdcmm::Parser, [770](#)
  - gdcmm::SequenceOfFragments, [950](#)
- GetBuffer2
  - gdcmm::Bitmap, [196](#)
- GetBufferAsRGBA
  - gdcmm::LookupTable, [651](#)
- GetBufferLength
  - gdcmm::Bitmap, [196](#)
  - gdcmm::JPEGLSCodec, [636](#)
  - gdcmm::PNMCodec, [827](#)
  - gdcmm::RLECodec, [910](#)
- GetBuildVersion
  - gdcmm::Version, [1249](#)
- GetByteValue
  - gdcmm::CSAElement, [278](#)
  - gdcmm::DataElement, [310](#)
- GetCSADataInfo
  - gdcmm::CSAHeader, [286](#)
- GetCSAEEnd
  - gdcmm::CSAHeader, [286](#)
- GetCSAElementByName
  - gdcmm::CSAHeader, [287](#)
- GetCSAHeaderDict
  - gdcmm::Dicts, [374](#)
- GetCSAHeaderDictEntry
  - gdcmm::CSAHeaderDict, [292](#)
- GetCSAImageHeaderInfoTag
  - gdcmm::CSAHeader, [287](#)
- GetCSASeriesHeaderInfoTag
  - gdcmm::CSAHeader, [287](#)
- GetCTImageSeriesUIDs
  - gdcmm::DirectoryHelper, [386](#)

- GetCWD
  - gdcm::System, [1078](#)
- GetCalledAETitle
  - gdcm::ServiceClassUser, [973](#)
  - gdcm::network::AAssociateRQPDU, [99](#)
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetCalledComputerName
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetCalledIPAddress
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetCalledIPPort
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetCallingAETitle
  - gdcm::network::AAssociateRQPDU, [99](#)
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetCenterOfRotation
  - gdcm::Surface, [1050](#)
- GetCharacterFromCurrentLocale
  - gdcm::QueryFactory, [875](#)
- GetCheckFileMetaInformation
  - gdcm::Writer, [1397](#)
- GetCipherType
  - gdcm::CAPICryptographicMessageSyntax, [233](#)
  - gdcm::CryptographicMessageSyntax, [274](#)
  - gdcm::OpenSSLCryptographicMessageSyntax, [746](#)
  - gdcm::OpenSSLP7CryptographicMessageSyntax, [751](#)
- GetCodec
  - gdcm::FileChangeTransferSyntax, [450](#)
- GetColorLevel
  - vtkImageColorViewer, [1345](#)
- GetColorWindow
  - vtkImageColorViewer, [1346](#)
- GetColumns
  - gdcm::Bitmap, [197](#)
  - gdcm::Overlay, [760](#)
- GetCommand
  - gdcm::Subject, [1044](#)
- GetConnectionInfo
  - gdcm::network::ULConnection, [1205](#)
- GetConstructorString
  - gdcm::Dicts, [374](#)
- GetContourReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1381](#)
- GetContourReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1382](#)
- GetCryptographicMessageSyntax
  - gdcm::Anonymizer, [115](#)
- GetCurrentByteIndex
  - gdcm::Parser, [770](#)
- GetCurrentDateTime
  - gdcm::System, [1077](#)
- GetCurrentModuleFileName
  - gdcm::System, [1077](#)
- GetCurrentProcessFileName
  - gdcm::System, [1078](#)
- GetCurrentResourcesDirectory
  - gdcm::System, [1078](#)
- GetCurve
  - gdcm::Pixmap, [809](#), [810](#)
- GetCurveDataDescriptor
  - gdcm::Curve, [302](#)
- GetDEEnd
  - gdcm::DataSet, [331](#)
- GetDES
  - gdcm::DataSet, [331](#)
- GetData
  - gdcm::DataEvent, [324](#)
- GetDataElement
  - gdcm::Bitmap, [197](#)
  - gdcm::DataSet, [330](#), [331](#)
  - gdcm::Item, [603](#)
- GetDataExtraRoot
  - gdcm::Testing, [1106](#)
- GetDataLength
  - gdcm::DataEvent, [324](#)
- GetDataRoot
  - gdcm::Testing, [1107](#)
- GetDataSet
  - gdcm::CSAHeader, [287](#)
  - gdcm::DataSetEvent, [340](#)
  - gdcm::File, [442](#)
- GetDataSetPos
  - gdcm::network::ULEvent, [1222](#)
- GetDataSetTransferSyntax
  - gdcm::FileMetaInformation, [466](#)
- GetDataSets
  - gdcm::network::ULBasicCallback, [1202](#)
- GetDataValueRepresentation
  - gdcm::Curve, [302](#)
- GetDebugFlag
  - gdcm::Trace, [1114](#)
- GetDebugStream
  - gdcm::Trace, [1114](#)
- GetDecodeLength
  - gdcm::Base64, [168](#)
- GetDefaultTransferSyntax
  - gdcm::PresentationContextGenerator, [842](#)
- GetDefs
  - gdcm::Global, [503](#)
  - gdcm::TableReader, [1086](#)
- GetDescription
  - gdcm::CSAHeaderDictEntry, [294](#)
  - gdcm::Exception, [431](#)
  - gdcm::ModuleEntry, [699](#)
  - gdcm::Overlay, [760](#)
- GetDescriptiveName
  - vtkGDCMImageReader, [1276](#)



- vtkGDCMImageReader2, [1291](#)
- vtkGDCMImageWriter, [1305](#)
- GetDict
  - gdcm::XMLDictReader, [1402](#)
- GetDictEntry
  - gdcm::Dict, [358](#)
  - gdcm::Dicts, [374](#)
  - gdcm::PrivateDict, [858](#)
- GetDictEntryByKeyword
  - gdcm::Dict, [359](#)
- GetDictEntryByName
  - gdcm::Dict, [359](#)
- GetDictName
  - gdcm::DictConverter, [363](#)
- GetDictVM
  - gdcm::Attribute, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [154](#)
- GetDictVR
  - gdcm::Attribute, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [154](#)
- GetDicts
  - gdcm::Global, [503](#), [504](#)
- GetDimension
  - gdcm::Bitmap, [197](#)
- GetDimensions
  - gdcm::Bitmap, [197](#)
  - gdcm::Curve, [303](#)
  - gdcm::ImageCodec, [545](#)
- GetDimensionsValue
  - gdcm::ImageHelper, [559](#)
- GetDimensionsValueForResolution
  - gdcm::StreamImageReader, [1013](#)
- GetDirectionCosines
  - gdcm::Image, [519](#)
- GetDirectionCosinesFromDataSet
  - gdcm::ImageHelper, [559](#)
- GetDirectionCosinesTolerance
  - gdcm::IPPSorter, [598](#)
- GetDirectionCosinesValue
  - gdcm::ImageHelper, [559](#)
- GetDirectories
  - gdcm::Directory, [384](#)
- GetElapsedTime
  - gdcm::network::ARTIMTimer, [129](#)
- GetElement
  - gdcm::Tag, [1093](#)
- GetElementTag
  - gdcm::Tag, [1093](#)
- GetEncodeLength
  - gdcm::Base64, [168](#)
- GetErrorCode
  - gdcm::Parser, [770](#)
- GetErrorFlag
  - gdcm::Trace, [1114](#)
- GetErrorStream
  - gdcm::Trace, [1114](#)
- GetErrorString
  - gdcm::Parser, [771](#)
- GetEvent
  - gdcm::network::ULEvent, [1222](#)
- GetEventName
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::DataEvent, [324](#)
  - gdcm::DataSetEvent, [340](#)
  - gdcm::Event, [428](#)
  - gdcm::FileNameEvent, [478](#)
  - gdcm::ProgressEvent, [865](#)
- GetExtension
  - gdcm::Filename, [473](#)
- GetFactoryInstance
  - gdcm::CryptoFactory, [271](#)
- GetFile
  - gdcm::Anonymizer, [115](#)
  - gdcm::DICOMDIRGenerator, [355](#)
  - gdcm::FileDecompressLookupTable, [454](#)
  - gdcm::FileDerivation, [457](#)
  - gdcm::FileExplicitFilter, [460](#)
  - gdcm::IconImageFilter, [510](#)
  - gdcm::PythonFilter, [870](#)
  - gdcm::Reader, [893](#)
  - gdcm::SplitMosaicFilter, [1008](#)
  - gdcm::StreamImageReader, [1013](#)
  - gdcm::StringFilter, [1039](#)
  - gdcm::Writer, [1397](#)
  - vtkGDCMMedicalImageProperties, [1313](#)
- GetFileExtensions
  - vtkGDCMImageReader, [1276](#)
  - vtkGDCMImageReader2, [1291](#)
  - vtkGDCMImageWriter, [1305](#)
- GetFileMetaInformationVersion
  - gdcm::FileMetaInformation, [466](#)
- GetFileName
  - gdcm::FileNameEvent, [478](#)
  - gdcm::Filename, [473](#)
  - gdcm::Testing, [1107](#)
  - vtkGDCMImageWriter, [1305](#)
  - vtkGDCMThreadedImageReader2, [1335](#)
- GetFileNames
  - gdcm::Testing, [1107](#)
- GetFilename
  - gdcm::FilenameGenerator, [481](#)
  - gdcm::TableReader, [1086](#)

GetFilenameFromTagToValue  
  gdcmm::Scanner, 920  
  gdcmm::StrictScanner, 1028

GetFileNames  
  gdcmm::Directory, 384  
  gdcmm::FilenameGenerator, 481  
  gdcmm::Scanner, 921  
  gdcmm::Sorter, 1000  
  gdcmm::StrictScanner, 1028

GetFileNamesFromSeriesUIDs  
  gdcmm::DirectoryHelper, 386

GetFiles  
  gdcmm::FileSet, 484

GetFiniteVolume  
  gdcmm::Surface, 1050

GetFirstSingleSerieUIDFileSet  
  gdcmm::SerieHelper, 966

GetForcePixelSpacing  
  gdcmm::ImageHelper, 559

GetForceRescaleInterceptSlope  
  gdcmm::ImageHelper, 559

GetFormat  
  gdcmm::CSAHeader, 288

GetFragBuffer  
  gdcmm::SequenceOfFragments, 950

GetFragment  
  gdcmm::SequenceOfFragments, 950

GetFragmentSizeMax  
  gdcmm::ImageFragmentSplitter, 556

GetFrameOfReference  
  gdcmm::DirectoryHelper, 387

GetFullLength  
  gdcmm::FileMetaInformation, 466

GetGDCMDataRoot  
  vtkGDCMTesting, 1327

GetGDCMImplementationClassUID  
  gdcmm::FileMetaInformation, 466

GetGDCMImplementationVersionName  
  gdcmm::FileMetaInformation, 466

GetGDCMSourceApplicationEntityTitle  
  gdcmm::FileMetaInformation, 467

GetGDCMUID  
  gdcmm::UIDGenerator, 1133

GetGroup  
  gdcmm::Curve, 303  
  gdcmm::Overlay, 760  
  gdcmm::Tag, 1094

GetHasExpired  
  gdcmm::network::ARTIMTimer, 130

GetHeader  
  gdcmm::File, 442

GetHeaderInfo  
  gdcmm::ImageCodec, 546  
  gdcmm::JPEG12Codec, 609

  gdcmm::JPEG16Codec, 612  
  gdcmm::JPEG2000Codec, 617  
  gdcmm::JPEG8Codec, 622  
  gdcmm::JPEGCodec, 628  
  gdcmm::JPEGLSCodec, 636  
  gdcmm::PGXCodec, 794  
  gdcmm::PNMCodec, 827  
  gdcmm::RAWCodec, 889  
  gdcmm::RLECodec, 910

GetHierarchicalSearchTags  
  gdcmm::QueryBase, 873  
  gdcmm::QueryImage, 877  
  gdcmm::QueryPatient, 879  
  gdcmm::QuerySeries, 882  
  gdcmm::QueryStudy, 884

GetHighBit  
  gdcmm::PixelFormat, 802

GetHostName  
  gdcmm::System, 1078

GetIODEntry  
  gdcmm::IOD, 588

GetIODFromFile  
  gdcmm::Defs, 346

GetIODFromSOPClassUID  
  gdcmm::SOPClassUIDToIOD, 997

GetIODNameFromMediaStorage  
  gdcmm::Defs, 346

GetIODs  
  gdcmm::Defs, 346

GetIOD  
  gdcmm::IODs, 595  
  gdcmm::SOPClassUIDToIOD, 997

GetIStream  
  gdcmm::network::ULEvent, 1222

GetIconImage  
  gdcmm::IconImageFilter, 510  
  gdcmm::IconImageGenerator, 513  
  gdcmm::Pixmap, 810  
  vtkGDCMImageReader, 1277  
  vtkGDCMImageReader2, 1291

GetIconImagePort  
  vtkGDCMImageReader2, 1291

GetIE  
  gdcmm::IODEntry, 591

GetImage  
  gdcmm::ImageReader, 566, 567  
  gdcmm::ImageWriter, 577, 578  
  gdcmm::PixmapWriter, 823  
  gdcmm::SplitMosaicFilter, 1008

GetImplementationClassUID  
  gdcmm::FileMetaInformation, 467

GetImplementationVersionName  
  gdcmm::FileMetaInformation, 467

GetIndex



- gdcm::SwapCode, [1071](#)
- gdcm::VM, [1258](#)
- GetInitialized
  - gdcm::CAPICryptographicMessageSyntax, [233](#)
- GetInput
  - gdcm::ImageToImageFilter, [574](#)
  - gdcm::PixmapToPixmapFilter, [819](#)
  - vtkImageColorViewer, [1346](#)
- GetInputFilename
  - gdcm::DictConverter, [363](#)
- GetInstance
  - gdcm::Global, [504](#)
- GetIntercept
  - gdcm::Image, [519](#)
  - gdcm::Rescaler, [904](#)
- GetInterfile
  - gdcm::CSAHeader, [288](#)
- GetInternal
  - gdcm::Preamble, [830](#)
- GetIsCommand
  - gdcm::network::PresentationDataValue, [849](#)
- GetIsLastFragment
  - gdcm::network::PresentationDataValue, [849](#)
- GetItem
  - gdcm::SequenceOfItems, [958](#)
- GetKey
  - gdcm::CSAElement, [278](#)
- GetKeys
  - gdcm::Scanner, [921](#)
  - gdcm::StrictScanner, [1028](#)
- GetKeyword
  - gdcm::DictEntry, [366](#)
- GetKeywordFromTag
  - gdcm::Dict, [359](#)
- GetLUTDescriptor
  - gdcm::LookupTable, [652](#)
- GetLUTLength
  - gdcm::LookupTable, [652](#)
- GetLUT
  - gdcm::Bitmap, [198](#)
  - gdcm::ImageCodec, [546](#)
  - gdcm::ImageHelper, [560](#)
  - gdcm::LookupTable, [652](#)
- GetLabel
  - gdcm::Orientation, [754](#)
- GetLastElement
  - gdcm::ParseException, [767](#)
- GetLastSystemError
  - gdcm::System, [1078](#)
- GetLength
  - gdcm::ByteValue, [224](#)
  - gdcm::CP246ExplicitDataElement, [268](#)
  - gdcm::DataElement, [311](#)
  - gdcm::DataSet, [332](#)
  - gdcm::Element, [393](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [400](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [411](#)
  - gdcm::ExplicitDataElement, [435](#)
  - gdcm::ExplicitImplicitDataElement, [438](#)
  - gdcm::Fragment, [500](#)
  - gdcm::ImplicitDataElement, [584](#)
  - gdcm::Item, [603](#)
  - gdcm::Preamble, [830](#)
  - gdcm::SequenceOfFragments, [950](#)
  - gdcm::SequenceOfItems, [959](#)
  - gdcm::Tag, [1094](#)
  - gdcm::UNExplicitDataElement, [1228](#)
  - gdcm::UNExplicitImplicitDataElement, [1231](#)
  - gdcm::VR16ExplicitDataElement, [1269](#)
  - gdcm::Value, [1245](#)
  - gdcm::VL, [1252](#)
  - gdcm::VM, [1258](#)
  - gdcm::VR, [1264](#)
- GetLocaleCharset
  - gdcm::System, [1078](#)
- GetLossless
  - gdcm::JPEGCodec, [628](#)
  - gdcm::JPEGLSCodec, [636](#)
- GetLossyFlag
  - gdcm::ImageCodec, [546](#)
- GetLossyFlagFromFile
  - gdcm::Testing, [1107](#)
- GetMD5DataImage
  - gdcm::Testing, [1108](#)
- GetMD5DataImages
  - gdcm::Testing, [1108](#)
- GetMD5FromBrokenFile
  - gdcm::Testing, [1108](#)
- GetMD5FromFile
  - gdcm::Testing, [1108](#)
- GetMD5MetaImage
  - vtkGDCMTesting, [1327](#)
- GetMHDMD5FromFile
  - vtkGDCMTesting, [1327](#)
- GetMPTType
  - gdcm::MeshPrimitive, [683](#)
- GetMPTTypeString
  - gdcm::MeshPrimitive, [683](#)
- GetMRImageSeriesUIDs
  - gdcm::DirectoryHelper, [387](#)
- GetMSString
  - gdcm::MediaStorage, [672](#)
- GetMSType
  - gdcm::MediaStorage, [672](#)
- GetMTime
  - vtkImageMapToColors16, [1359](#)
- GetMacro
  - gdcm::Macros, [661](#)

- GetMacroEntry
  - gdcm::Macro, [658](#)
- GetMacros
  - gdcm::Defs, [347](#)
- GetMajorAxisFromPatientRelativeDirectionCosine
  - gdcm::Orientation, [754](#)
- GetMajorVersion
  - gdcm::Version, [1250](#)
- GetManifold
  - gdcm::Surface, [1050](#)
- GetMapping
  - gdcm::Scanner, [921](#)
  - gdcm::StrictScanner, [1028](#)
- GetMappingFromTagToValue
  - gdcm::Scanner, [921](#)
  - gdcm::StrictScanner, [1028](#)
- GetMappings
  - gdcm::Scanner, [921](#)
  - gdcm::StrictScanner, [1029](#)
- GetMax
  - gdcm::PixelFormat, [802](#)
- GetMaxLength
  - gdcm::PersonName, [789](#)
- GetMaxPDULength
  - gdcm::network::ULConnectionInfo, [1212](#)
- GetMaxPDUSize
  - gdcm::network::ULConnection, [1205](#)
- GetMaximumLength
  - gdcm::network::MaximumLengthSub, [663](#)
- GetMaximumLengthSub
  - gdcm::network::UserInformation, [1239](#)
- GetMaximumPointDistance
  - gdcm::Surface, [1050](#)
- GetMeanPointDistance
  - gdcm::Surface, [1050](#)
- GetMediaStorage
  - gdcm::DataSet, [332](#)
  - gdcm::FileMetaInformation, [467](#)
- GetMediaStorageAsString
  - gdcm::FileMetaInformation, [467](#)
- GetMediaStorageDataFile
  - gdcm::Testing, [1108](#)
- GetMediaStorageDataFiles
  - gdcm::Testing, [1108](#)
- GetMediaStorageFromFile
  - gdcm::Testing, [1109](#)
- GetMeshPrimitive
  - gdcm::Surface, [1051](#)
- GetMessageHeader
  - gdcm::network::PresentationDataValue, [849](#)
- GetMetaInformationTS
  - gdcm::FileMetaInformation, [467](#)
- GetMin
  - gdcm::PixelFormat, [802](#)
- GetMinorVersion
  - gdcm::Version, [1250](#)
- GetModality
  - gdcm::MediaStorage, [671](#)
- GetModalityDimension
  - gdcm::MediaStorage, [672](#)
- GetModule
  - gdcm::Modules, [702](#)
- GetModuleEntry
  - gdcm::NestedModuleEntries, [723](#)
- GetModuleEntryInMacros
  - gdcm::Module, [695](#)
- GetModules
  - gdcm::Defs, [347](#)
- GetMrProtocol
  - gdcm::CSAHeader, [288](#)
- GetMrProtocolByName
  - gdcm::MrProtocol, [710](#)
- GetName
  - gdcm::CSAElement, [278](#)
  - gdcm::CSAHeaderDictEntry, [294](#)
  - gdcm::DictEntry, [367](#)
  - gdcm::Filename, [473](#)
  - gdcm::GroupDict, [507](#)
  - gdcm::IODEntry, [591](#)
  - gdcm::Macro, [659](#)
  - gdcm::Module, [695](#)
  - gdcm::ModuleEntry, [699](#)
  - gdcm::PDBElement, [777](#)
  - gdcm::QueryBase, [873](#)
  - gdcm::QueryImage, [877](#)
  - gdcm::QueryPatient, [879](#)
  - gdcm::QuerySeries, [882](#)
  - gdcm::QueryStudy, [884](#)
  - gdcm::UIDs, [1158](#)
  - gdcm::network::AbstractSyntax, [105](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::TransferSyntaxSub, [1124](#)
- GetNeedByteSwap
  - gdcm::Bitmap, [198](#)
  - gdcm::ImageCodec, [546](#)
- GetNegotiatedType
  - gdcm::TransferSyntax, [1121](#)
- GetNestedDataSet
  - gdcm::Item, [604](#)
- GetNextSingleSerieUIDFileSet
  - gdcm::SerieHelper, [966](#)
- GetNoOfItems
  - gdcm::CSAElement, [278](#)
- GetNumberOfComponents
  - gdcm::PersonName, [790](#)
- GetNumberOfContourReferencedFrameOfReferences
  - vtkRTStructSetProperties, [1382](#)
- GetNumberOfCurves

- gdcm::Curve, [303](#)
- gdcm::Pixmap, [810](#)
- GetNumberOfDimensions
  - gdcm::Bitmap, [198](#)
  - gdcm::ImageCodec, [546](#)
- GetNumberOfElementsFromArray
  - gdcm::VM, [1258](#)
- GetNumberOfFileNames
  - gdcm::Testing, [1109](#)
- GetNumberOfFileNames
  - gdcm::FilenameGenerator, [481](#)
- GetNumberOfFragments
  - gdcm::SequenceOfFragments, [951](#)
- GetNumberOfIODs
  - gdcm::IOD, [589](#)
- GetNumberOfIconImages
  - gdcm::IconImageFilter, [511](#)
- GetNumberOfItems
  - gdcm::SequenceOfItems, [959](#)
- GetNumberOfMD5DataImages
  - gdcm::Testing, [1109](#)
- GetNumberOfMD5MetaImages
  - vtkGDCMTesting, [1327](#)
- GetNumberOfMSSString
  - gdcm::MediaStorage, [672](#)
- GetNumberOfMSType
  - gdcm::MediaStorage, [673](#)
- GetNumberOfMediaStorageDataFiles
  - gdcm::Testing, [1109](#)
- GetNumberOfModality
  - gdcm::MediaStorage, [672](#)
- GetNumberOfModuleEntries
  - gdcm::NestedModuleEntries, [723](#)
- GetNumberOfOverlays
  - gdcm::Pixmap, [810](#)
- GetNumberOfPoints
  - gdcm::Curve, [303](#)
- GetNumberOfPresentationContext
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetNumberOfPresentationContextAC
  - gdcm::network::AAssociateACPDU, [92](#)
- GetNumberOfPresentationDataValues
  - gdcm::network::PDataTFPDU, [774](#)
- GetNumberOfPrimitivesData
  - gdcm::MeshPrimitive, [684](#)
- GetNumberOfReferencedFrameOfReferences
  - vtkRTStructSetProperties, [1382](#)
- GetNumberOfSOPClassToIOD
  - gdcm::SOPClassUIDToIOD, [997](#)
- GetNumberOfSegments
  - gdcm::SegmentWriter, [944](#)
- GetNumberOfStructureSetROIs
  - vtkRTStructSetProperties, [1382](#)
- GetNumberOfSurfacePoints
  - gdcm::Surface, [1051](#)
- GetNumberOfSurfaces
  - gdcm::SurfaceReader, [1065](#)
  - gdcm::SurfaceWriter, [1068](#)
- GetNumberOfTransferSyntaxStrings
  - gdcm::UIDs, [1158](#)
- GetNumberOfTransferSyntaxes
  - gdcm::PresentationContext, [835](#)
  - gdcm::network::PresentationContextRQ, [845](#)
- GetNumberOfValues
  - gdcm::Attribute, [137](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [154](#)
- GetNumberOfVectors
  - gdcm::Surface, [1051](#)
- GetObliquityThresholdCosineValue
  - gdcm::Orientation, [754](#)
- GetOffScreenRendering
  - vtkImageColorViewer, [1346](#)
- GetOptionalTags
  - gdcm::QueryBase, [873](#)
  - gdcm::QueryImage, [877](#)
  - gdcm::QueryPatient, [880](#)
  - gdcm::QuerySeries, [882](#)
  - gdcm::QueryStudy, [885](#)
- GetOrderedValues
  - gdcm::Scanner, [922](#)
  - gdcm::StrictScanner, [1029](#)
- GetOrigin
  - gdcm::Image, [519](#)
  - gdcm::Overlay, [760](#)
- GetOriginValue
  - gdcm::ImageHelper, [560](#)
- GetOutput
  - gdcm::ImageConverter, [554](#)
- GetOutput
  - gdcm::BitmapToBitmapFilter, [209](#)
  - gdcm::ImageToImageFilter, [574](#)
  - gdcm::PixmapToPixmapFilter, [819](#)
- GetOutputAsBitmap
  - gdcm::BitmapToBitmapFilter, [209](#)
- GetOutputAsPixmap
  - gdcm::PixmapToPixmapFilter, [819](#)
- GetOutputFilename
  - gdcm::DictConverter, [363](#)
- GetOutputType
  - gdcm::DictConverter, [363](#)
- GetOverlay
  - gdcm::Pixmap, [810](#), [811](#)
  - vtkGDCMImageReader, [1277](#)
  - vtkGDCMImageReader2, [1291](#)
- GetOverlayData

- gdcm::Overlay, [760](#)
- GetOverlayPort
  - vtkGDCMImageReader2, [1292](#)
- GetOverlayTypeAsString
  - gdcm::Overlay, [761](#)
- GetOverlayTypeFromString
  - gdcm::Overlay, [761](#)
- GetOverlayVisibility
  - vtkImageColorViewer, [1346](#)
- GetOwner
  - gdcm::PrivateTag, [861](#)
- GetPDBEEnd
  - gdcm::PDBHeader, [781](#)
- GetPDBElementByName
  - gdcm::PDBHeader, [781](#)
- GetPDBInfoTag
  - gdcm::PDBHeader, [781](#)
- GetPDUs
  - gdcm::network::ULEvent, [1222](#)
- GetPDVs
  - gdcm::network::PDUFactory, [788](#)
- GetPIString
  - gdcm::PhotometricInterpretation, [797](#)
- GetPIType
  - gdcm::PhotometricInterpretation, [797](#)
- GetPMSRescaleInterceptSlope
  - gdcm::ImageHelper, [560](#)
- GetPath
  - gdcm::Filename, [473](#)
- GetPattern
  - gdcm::FilenameGenerator, [481](#)
- GetPermissions
  - gdcm::System, [1079](#)
- GetPhotometricInterpretation
  - gdcm::Bitmap, [198](#)
  - gdcm::ImageChangePhotometricInterpretation, [529](#)
  - gdcm::ImageCodec, [546](#)
- GetPhotometricInterpretationValue
  - gdcm::ImageHelper, [560](#)
- GetPixelFormat
  - gdcm::Bitmap, [199](#)
  - gdcm::ImageCodec, [547](#)
- GetPixelFormatValue
  - gdcm::ImageHelper, [560](#)
- GetPixelRepresentation
  - gdcm::PixelFormat, [802](#)
- GetPixelSize
  - gdcm::PixelFormat, [803](#)
- GetPixelSpacingDataRoot
  - gdcm::Testing, [1109](#)
- GetPixmap
  - gdcm::FileDecompressLookupTable, [454](#)
  - gdcm::IconImageGenerator, [514](#)
  - gdcm::PixmapReader, [815](#)
  - gdcm::PixmapWriter, [823](#)
- GetPlanarConfiguration
  - gdcm::Bitmap, [199](#)
  - gdcm::ImageChangePlanarConfiguration, [532](#)
  - gdcm::ImageCodec, [547](#)
- GetPlanarConfigurationValue
  - gdcm::ImageHelper, [560](#)
- GetPointCoordinatesData
  - gdcm::Surface, [1051](#)
- GetPointPositionAccuracy
  - gdcm::Surface, [1051](#)
- GetPointer
  - gdcm::ByteValue, [225](#)
  - gdcm::LookupTable, [652](#)
  - gdcm::SmartPointer, [992](#)
  - vtkLookupTable16, [1376](#)
- GetPointerFromElement
  - gdcm::ImageHelper, [561](#)
- GetPointsBoundingBoxCoordinates
  - gdcm::Surface, [1052](#)
- GetPosition
  - vtkImageColorViewer, [1346](#)
- GetPreamble
  - gdcm::FileMetaInformation, [467](#), [468](#)
- GetPrefix
  - gdcm::FilenameGenerator, [482](#)
- GetPresentationContext
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextACByID
  - gdcm::network::ULConnection, [1205](#)
- GetPresentationContextAC
  - gdcm::network::AAssociateACPDU, [92](#)
- GetPresentationContextByAbstractSyntax
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextByID
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetPresentationContextIDFromPresentationContext
  - gdcm::network::ULConnection, [1205](#)
- GetPresentationContextID
  - gdcm::PresentationContext, [835](#)
  - gdcm::network::PresentationContextAC, [838](#)
  - gdcm::network::PresentationContextRQ, [845](#)
  - gdcm::network::PresentationDataValue, [849](#)
- GetPresentationContextRQByID
  - gdcm::network::ULConnection, [1206](#)
- GetPresentationContexts
  - gdcm::PresentationContextGenerator, [842](#)
  - gdcm::network::AAssociateRQPDU, [100](#)
  - gdcm::network::ULConnection, [1206](#)
- GetPresentationDataValue
  - gdcm::network::PDataTFPDU, [775](#)
- GetPrettyPrint
  - gdcm::JSON, [639](#)
- GetPrimitiveData

- gdcm::MeshPrimitive, [684](#)
- GetPrimitiveType
  - gdcm::MeshPrimitive, [685](#)
- GetPrimitivesData
  - gdcm::MeshPrimitive, [684](#)
- GetPrintStyle
  - gdcm::Printer, [854](#)
  - gdcm::XMLPrinter, [1404](#)
- GetPrivateCreator
  - gdcm::DataSet, [332](#)
  - gdcm::Tag, [1094](#)
- GetPrivateDict
  - gdcm::Dicts, [375](#)
  - gdcm::XMLPrivateDictReader, [1409](#)
- GetProcessingAlgorithm
  - gdcm::Surface, [1052](#)
- GetProgress
  - gdcm::ProgressEvent, [865](#)
- GetPropertyCategory
  - gdcm::Segment, [929](#)
- GetPropertyType
  - gdcm::Segment, [929](#), [930](#)
- GetPropertyTypeModifiers
  - gdcm::Segment, [930](#)
- GetProtocol
  - gdcm::network::ULConnection, [1206](#)
- GetPublicDict
  - gdcm::Dicts, [375](#)
- GetQuality
  - gdcm::JPEG2000Codec, [617](#)
  - gdcm::JPEGCodec, [628](#)
- GetQueryDataSet
  - gdcm::BaseQuery, [178](#), [179](#)
- GetQueryLevel
  - gdcm::QueryBase, [873](#)
  - gdcm::QueryImage, [877](#)
  - gdcm::QueryPatient, [880](#)
  - gdcm::QuerySeries, [882](#)
  - gdcm::QueryStudy, [885](#)
- GetQueryLevelFromQueryRoot
  - gdcm::BaseRootQuery, [184](#)
- GetQueryLevelFromString
  - gdcm::BaseRootQuery, [184](#)
- GetQueryLevelString
  - gdcm::BaseRootQuery, [184](#)
- GetRAWMD5FromFile
  - vtkGDCMTesting, [1328](#)
- GetRTStructSeriesUIDs
  - gdcm::DirectoryHelper, [387](#)
- GetRate
  - gdcm::JPEG2000Codec, [618](#)
- GetRealWorldValueMappingContent
  - gdcm::ImageHelper, [561](#)
- GetReason
  - gdcm::network::PresentationContextAC, [838](#)
- GetRecommendedDisplayCIELabValue
  - gdcm::Surface, [1052](#)
- GetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, [1052](#)
- GetRecommendedPresentationOpacity
  - gdcm::Surface, [1053](#)
- GetRecommendedPresentationType
  - gdcm::Surface, [1053](#)
- GetRef
  - gdcm::IODEntry, [591](#)
- GetReferencedFrameOfReferenceClassUID
  - vtkRTStructSetProperties, [1382](#)
- GetReferencedFrameOfReferenceInstanceUID
  - vtkRTStructSetProperties, [1383](#)
- GetRegion
  - gdcm::ImageRegionReader, [571](#)
- GetRequiredDataSet
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [688](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [691](#)
- GetRequiredTags
  - gdcm::QueryBase, [873](#)
  - gdcm::QueryImage, [878](#)
  - gdcm::QueryPatient, [880](#)
  - gdcm::QuerySeries, [882](#)
  - gdcm::QueryStudy, [885](#)
- GetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, [561](#)
- GetReserved43\_74
  - gdcm::network::AAssociateRQPDU, [100](#)
- GetResponses
  - gdcm::network::ULBasicCallback, [1202](#)
- GetRetired
  - gdcm::DictEntry, [367](#)
- GetRoot
  - gdcm::UIDGenerator, [1133](#)
- GetRows
  - gdcm::Bitmap, [199](#)
  - gdcm::Overlay, [761](#)
- GetSOPClassUIDFromIOD
  - gdcm::SOPClassUIDToIOD, [997](#)
- GetSOPClassUIDToIODs
  - gdcm::SOPClassUIDToIOD, [997](#)
- GetSOPClassUIDToIOD
  - gdcm::SOPClassUIDToIOD, [997](#)
- GetSOPClassUID
  - gdcm::DirectoryHelper, [387](#)
- GetSOPInstanceUID
  - gdcm::BaseQuery, [179](#)
- GetSTATESString
  - gdcm::Surface, [1053](#)
- GetSTATES

- gdcm::Surface, [1053](#)
- GetSamplesPerPixel
  - gdcm::PhotometricInterpretation, [797](#)
  - gdcm::PixelFormat, [803](#)
- GetScalarType
  - gdcm::PixelFormat, [803](#)
- GetScalarTypeAsString
  - gdcm::PixelFormat, [803](#)
- GetScanner
  - gdcm::DICOMDIRGenerator, [355](#)
- GetSegment
  - gdcm::SegmentWriter, [944](#)
- GetSegmentAlgorithmName
  - gdcm::Segment, [930](#)
- GetSegmentAlgorithmType
  - gdcm::Segment, [930](#)
- GetSegmentDescription
  - gdcm::Segment, [930](#)
- GetSegmentLabel
  - gdcm::Segment, [930](#)
- GetSegmentNumber
  - gdcm::Segment, [931](#)
- GetSegments
  - gdcm::SegmentReader, [940](#)
  - gdcm::SegmentWriter, [944](#)
- GetSelectedPrivateGroupOffsetFromFile
  - gdcm::Testing, [1109](#)
- GetSelectedTagsOffsetFromFile
  - gdcm::Testing, [1110](#)
- GetSequenceOfFragments
  - gdcm::DataElement, [311](#)
- GetSeriesUIDsBySOPClassUID
  - gdcm::DirectoryHelper, [387](#)
- GetSize
  - gdcm::VR, [1264](#)
  - vtkImageColorViewer, [1346](#)
- GetSizeof
  - gdcm::VR, [1264](#)
- GetSliceArray
  - gdcm::MrProtocol, [710](#)
- GetSliceMax
  - vtkImageColorViewer, [1346](#)
- GetSliceMin
  - vtkImageColorViewer, [1347](#)
- GetSliceRange
  - vtkImageColorViewer, [1347](#)
- GetSlope
  - gdcm::Image, [520](#)
  - gdcm::Rescaler, [904](#)
- GetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [468](#)
- GetSourceDirectory
  - gdcm::Testing, [1110](#)
- GetSpacing
  - gdcm::Image, [520](#)
- GetSpacingTagFromMediaStorage
  - gdcm::ImageHelper, [561](#)
- GetSpacingValue
  - gdcm::ImageHelper, [561](#)
- GetStart
  - gdcm::ByteBuffer, [216](#)
- GetState
  - gdcm::network::ULConnection, [1206](#)
- GetStateIndex
  - gdcm::network, [83](#)
- GetStream
  - gdcm::Trace, [1115](#)
- GetStreamCurrentPosition
  - gdcm::Reader, [894](#)
- GetStreamOffsetFromFile
  - gdcm::Testing, [1110](#)
- GetStreamPtr
  - gdcm::Reader, [894](#)
  - gdcm::Writer, [1397](#)
- GetString
  - gdcm::MediaStorage, [673](#)
  - gdcm::PhotometricInterpretation, [797](#)
  - gdcm::TransferSyntax, [1121](#)
  - gdcm::UIDs, [1159](#)
- GetStringValueFromTag
  - gdcm::DirectoryHelper, [388](#)
- GetStructureSetObservationNumber
  - vtkRTStructSetProperties, [1383](#)
- GetStructureSetROIDescription
  - vtkRTStructSetProperties, [1383](#)
- GetStructureSetROIGenerationAlgorithm
  - vtkRTStructSetProperties, [1383](#)
- GetStructureSetROIName
  - vtkRTStructSetProperties, [1383](#)
- GetStructureSetROINumber
  - vtkRTStructSetProperties, [1383](#)
- GetStructureSetROIObservationLabel
  - vtkRTStructSetProperties, [1384](#)
- GetStructureSetROIRefFrameRefUID
  - vtkRTStructSetProperties, [1384](#)
- GetStructureSetRTROIInterpretedType
  - vtkRTStructSetProperties, [1384](#)
- GetSurface
  - gdcm::Segment, [931](#)
- GetSurfaceComments
  - gdcm::Surface, [1053](#)
- GetSurfaceCount
  - gdcm::Segment, [931](#)
- GetSurfaceNumber
  - gdcm::Surface, [1053](#)
- GetSurfaceProcessing
  - gdcm::Surface, [1053](#)
- GetSurfaceProcessingDescription



- gdcmm::Surface, [1054](#)
- GetSurfaceProcessingRatio
  - gdcmm::Surface, [1054](#)
- GetSurfaces
  - gdcmm::Segment, [931](#)
- GetSwapCode
  - gdcmm::TransferSyntax, [1121](#)
- GetSwapCodeString
  - gdcmm::SwapCode, [1072](#)
- GetSyngoDT
  - gdcmm::CSAElement, [278](#)
- GetTSString
  - gdcmm::TransferSyntax, [1121](#)
- GetTSType
  - gdcmm::TransferSyntax, [1122](#)
- GetTable
  - gdcmm::SequenceOfFragments, [951](#)
- GetTableEntry
  - gdcmm::Table, [1083](#)
- GetTag
  - gdcmm::AnonymizeEvent, [109](#)
  - gdcmm::Attribute, [137](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcmm::DataElement, [312](#)
- GetTagListByLevel
  - gdcmm::BaseRootQuery, [184](#)
  - gdcmm::FindPatientRootQuery, [494](#)
  - gdcmm::FindStudyRootQuery, [497](#)
  - gdcmm::MovePatientRootQuery, [705](#)
  - gdcmm::MoveStudyRootQuery, [708](#)
  - gdcmm::WLMFindQuery, [1392](#)
- GetTempDirectory
  - gdcmm::Testing, [1110](#)
- GetTempDirectoryW
  - gdcmm::Testing, [1110](#)
- GetTempFilename
  - gdcmm::Testing, [1111](#)
- GetTempFilenameW
  - gdcmm::Testing, [1111](#)
- GetTimeout
  - gdcmm::ServiceClassUser, [973](#)
  - gdcmm::network::ARTIMTimer, [130](#)
- GetTimer
  - gdcmm::network::ULConnection, [1206](#)
- GetTimezoneOffsetFromUTC
  - gdcmm::System, [1079](#)
- GetToplevel
  - gdcmm::Directory, [384](#)
- GetTransferSyntax
  - gdcmm::Bitmap, [199](#)
  - gdcmm::ImageChangeTransferSyntax, [537](#)
  - gdcmm::PresentationContext, [835](#)
  - gdcmm::network::PresentationContextAC, [838](#)
  - gdcmm::network::PresentationContextRQ, [845](#)
- GetTransferSyntaxString
  - gdcmm::UIDs, [1159](#)
- GetTransferSyntaxStrings
  - gdcmm::UIDs, [1159](#)
- GetTransferSyntaxes
  - gdcmm::network::PresentationContextRQ, [846](#)
- GetType
  - gdcmm::ModuleEntry, [699](#)
  - gdcmm::Orientation, [754](#)
  - gdcmm::Overlay, [761](#)
  - gdcmm::PhotometricInterpretation, [797](#)
- GetTypeAsEnum
  - gdcmm::Overlay, [761](#)
- GetTypeFromTag
  - gdcmm::Defs, [347](#)
  - gdcmm::IOD, [589](#)
- GetTypeOfData
  - gdcmm::Curve, [303](#)
- GetTypeOfDataDescription
  - gdcmm::Curve, [303](#)
- GetTypeString
  - gdcmm::Type, [1130](#)
- GetTypeType
  - gdcmm::Type, [1130](#)
- GetUIDName
  - gdcmm::UIDs, [1159](#)
- GetUIDString
  - gdcmm::UIDs, [1159](#)
- GetUniqueTags
  - gdcmm::QueryBase, [874](#)
  - gdcmm::QueryImage, [878](#)
  - gdcmm::QueryPatient, [880](#)
  - gdcmm::QuerySeries, [883](#)
  - gdcmm::QueryStudy, [885](#)
- GetUnpackBuffer
  - gdcmm::Overlay, [761](#)
- GetUnpackBufferLength
  - gdcmm::Overlay, [762](#)
- GetUsage
  - gdcmm::IODEntry, [591](#)
- GetUsageString
  - gdcmm::Usage, [1235](#)
- GetUsageType
  - gdcmm::IODEntry, [591](#)
  - gdcmm::Usage, [1235](#)
- GetUserData
  - gdcmm::Parser, [771](#)
- GetUserInformation
  - gdcmm::network::AAssociateACPDU, [92](#)
  - gdcmm::network::AAssociateRQPDU, [101](#)
- GetVIEWType

- gdcmm::Surface, [1054](#)
- GetVIEWTypeString
  - gdcmm::Surface, [1055](#)
- GetVL16Max
  - gdcmm::VL, [1252](#)
- GetVL32Max
  - gdcmm::VL, [1253](#)
- GetVMString
  - gdcmm::VM, [1258](#)
- GetVMType
  - gdcmm::VM, [1259](#)
- GetVMTypeFromLength
  - gdcmm::VM, [1259](#)
- GetVRFromTag
  - gdcmm, [63](#)
- GetVRString
  - gdcmm::VR, [1264](#)
- GetVRStringFromFile
  - gdcmm::VR, [1265](#)
- GetVRType
  - gdcmm::VR, [1265](#)
- GetVRTypeFromFile
  - gdcmm::VR, [1265](#)
- GetVTKDataRoot
  - vtkGDCMTesting, [1328](#)
- GetValidDataSet
  - gdcmm::WLMFindQuery, [1392](#)
- GetValidatedFile
  - gdcmm::Validate, [1242](#)
- GetValue
  - gdcmm::Attribute, [138](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [145](#), [146](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcmm::CSAElement, [279](#)
  - gdcmm::DataElement, [312](#)
  - gdcmm::Element, [394](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [400](#)
  - gdcmm::PDBelement, [778](#)
  - gdcmm::Scanner, [922](#)
  - gdcmm::StrictScanner, [1029](#)
- GetValueAsSQ
  - gdcmm::DataElement, [313](#)
- GetValues
  - gdcmm::Attribute, [138](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [146](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcmm::Element, [394](#)
  - gdcmm::Scanner, [922](#)
  - gdcmm::StrictScanner, [1029](#)
- GetVectorAccuracy
  - gdcmm::Surface, [1054](#)
- GetVectorCoordinateData
  - gdcmm::Surface, [1054](#)
- GetVectorDimensionality
  - gdcmm::Surface, [1054](#)
- GetVersion
  - gdcmm::MrProtocol, [711](#)
  - gdcmm::Version, [1250](#)
- GetVL
  - gdcmm::DataElement, [313](#)
- GetVM
  - gdcmm::Attribute, [138](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [146](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_3 >, [150](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_8 >, [151](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM2↵\_2n >, [159](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM2\_n >, [161](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM3↵\_3n >, [162](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM3\_n >, [164](#)
  - gdcmm::CSAElement, [279](#)
  - gdcmm::CSAHeaderDictEntry, [294](#)
  - gdcmm::DictEntry, [367](#)
  - gdcmm::Element, [394](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [400](#)
- GetVR
  - gdcmm::Attribute, [138](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [146](#)
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, [155](#)
  - gdcmm::CSAElement, [279](#)
  - gdcmm::CSAHeaderDictEntry, [294](#)
  - gdcmm::DataElement, [313](#)
  - gdcmm::DictEntry, [367](#)
  - gdcmm::Element, [394](#)
  - gdcmm::Element< TVR, VM::VM1\_n >, [400](#)
- GetWarningFlag
  - gdcmm::Trace, [1115](#)
- GetWarningStream
  - gdcmm::Trace, [1115](#)
- GetWindowName
  - vtkImageColorViewer, [1347](#)
- GetXMax
  - gdcmm::BoxRegion, [213](#)
- GetXMin



- gdcmm::BoxRegion, [213](#)
- GetYMax
  - gdcmm::BoxRegion, [214](#)
- GetYMin
  - gdcmm::BoxRegion, [214](#)
- GetZMax
  - gdcmm::BoxRegion, [214](#)
- GetZMin
  - gdcmm::BoxRegion, [214](#)
- GetZSpacing
  - gdcmm::IPPSorter, [598](#)
- GetZSpacingTagFromMediaStorage
  - gdcmm::ImageHelper, [562](#)
- GetZSpacingTolerance
  - gdcmm::IPPSorter, [598](#)
- Global
  - gdcmm::Defs, [348](#)
  - gdcmm::Dicts, [376](#)
  - gdcmm::Global, [502](#)
- GlobalInstance
  - gdcmm, [77](#)
- GrabOverlayFromPixelData
  - gdcmm::Overlay, [762](#)
- group
  - gdcmm::SerieHelper::Rule, [914](#)
- GroupDict
  - gdcmm::GroupDict, [506](#)
- GroupStringVector
  - gdcmm::GroupDict, [506](#)
- GuessFromModality
  - gdcmm::MediaStorage, [673](#)
- HandleBulkData
  - gdcmm::XMLPrinter, [1404](#)
- HandleDataSet
  - gdcmm::network::ULBasicCallback, [1202](#)
  - gdcmm::network::ULConnectionCallback, [1210](#)
  - gdcmm::network::ULWritingCallback, [1226](#)
- HandleDescription
  - gdcmm::XMLDictReader, [1402](#)
  - gdcmm::XMLPrivateDictReader, [1409](#)
- HandleEntry
  - gdcmm::XMLDictReader, [1402](#)
  - gdcmm::XMLPrivateDictReader, [1409](#)
- HandleEvent
  - gdcmm::network::ULTransitionTable, [1224](#)
- HandleIODEntry
  - gdcmm::TableReader, [1086](#)
- HandleIOD
  - gdcmm::TableReader, [1086](#)
- HandleMacro
  - gdcmm::TableReader, [1087](#)
- HandleMacroEntry
  - gdcmm::TableReader, [1087](#)
- HandleMacroEntryDescription
  - gdcmm::TableReader, [1087](#)
- HandleModule
  - gdcmm::TableReader, [1087](#)
- HandleModuleEntry
  - gdcmm::TableReader, [1087](#)
- HandleModuleEntryDescription
  - gdcmm::TableReader, [1087](#)
- HandleModuleInclude
  - gdcmm::TableReader, [1088](#)
- HandleResponse
  - gdcmm::network::ULBasicCallback, [1203](#)
  - gdcmm::network::ULConnectionCallback, [1210](#)
  - gdcmm::network::ULWritingCallback, [1226](#)
- HasObserver
  - gdcmm::Subject, [1044](#)
- IODEntry
  - gdcmm::IODEntry, [591](#)
- IODMapType
  - gdcmm::IODs, [593](#)
- IODMapTypeConstIterator
  - gdcmm::IODs, [594](#)
- IODName
  - gdcmm::IODs, [594](#)
- IODs
  - gdcmm::IODs, [594](#)
- IOD
  - gdcmm::IOD, [588](#)
- IPPSorter
  - gdcmm::IPPSorter, [597](#)
- Icon
  - gdcmm::Pixmap, [812](#)
- IconDataScalarType
  - vtkGDCMImageReader, [1285](#)
  - vtkGDCMImageReader2, [1300](#)
- IconImage
  - gdcmm, [59](#)
- IconImageDataExtent
  - vtkGDCMImageReader, [1285](#)
  - vtkGDCMImageReader2, [1300](#)
- IconImageFilter
  - gdcmm::IconImageFilter, [509](#)
- IconImageGenerator
  - gdcmm::IconImageGenerator, [512](#)
- IconNumberOfScalarComponents
  - vtkGDCMImageReader, [1285](#)
  - vtkGDCMImageReader2, [1300](#)
- ID
  - gdcmm::PresentationContext, [836](#)
- ignore\_char
  - gdcmm::ignore\_char, [515](#)
- Image
  - gdcmm::Image, [518](#)

- ImageActor
  - vtkImageColorViewer, [1355](#)
- ImageApplyLookupTable
  - gdcm::ImageApplyLookupTable, [525](#)
- ImageChangePhotometricInterpretation
  - gdcm::ImageChangePhotometricInterpretation, [528](#)
  - gdcm::ImageCodec, [551](#)
- ImageChangePlanarConfiguration
  - gdcm::ImageChangePlanarConfiguration, [532](#)
- ImageChangeTransferSyntax
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageChangeTransferSyntax, [536](#)
- ImageCodec
  - gdcm::ImageCodec, [542](#)
- ImageConverter
  - gdcm::ImageConverter, [553](#)
- ImageFormat
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1300](#)
- ImageFragmentSplitter
  - gdcm::ImageFragmentSplitter, [556](#)
- ImageOrientationPatient
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1300](#)
- ImagePositionPatient
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1301](#)
- ImagePositionPatientOrdering
  - gdcm::SerieHelper, [966](#)
- ImageReader
  - gdcm::ImageReader, [566](#)
- ImageRegionReader
  - gdcm::ImageRegionReader, [570](#)
  - gdcm::JPEG2000Codec, [620](#)
  - gdcm::JPEGCodec, [631](#)
  - gdcm::JPEGLSCodec, [638](#)
  - gdcm::RLECodec, [912](#)
- ImageToImageFilter
  - gdcm::ImageToImageFilter, [574](#)
- ImageWriter
  - gdcm::ImageWriter, [577](#)
- ImplementationClassUIDSub
  - gdcm::network::ImplementationClassUIDSub, [579](#)
- ImplementationUIDSub
  - gdcm::network::ImplementationUIDSub, [580](#)
- ImplementationVersionNameSub
  - gdcm::network::ImplementationVersionNameSub, [581](#)
- IncompleteLUT
  - gdcm::LookupTable, [655](#)
- InitFromRQ
  - gdcm::network::AAssociateACPDU, [92](#)
- InitOpenSSL
  - gdcm::OpenSSLCryptoFactory, [743](#)
- Initialize
  - gdcm::network::ULConnectionInfo, [1212](#)
- InitializeBlueLUT
  - gdcm::LookupTable, [652](#)
- InitializeConnection
  - gdcm::ServiceClassUser, [973](#)
  - gdcm::network::ULConnection, [1206](#)
- InitializeDataSet
  - gdcm::BaseRootQuery, [184](#)
  - gdcm::FindPatientRootQuery, [494](#)
  - gdcm::FindStudyRootQuery, [497](#)
  - gdcm::MovePatientRootQuery, [705](#)
  - gdcm::MoveStudyRootQuery, [708](#)
  - gdcm::WLMFindQuery, [1392](#)
- InitializeGreenLUT
  - gdcm::LookupTable, [653](#)
- InitializeIncomingConnection
  - gdcm::network::ULConnection, [1207](#)
- InitializeLUT
  - gdcm::LookupTable, [653](#)
- InitializeRTStructSet
  - vtkGDCMPolyDataWriter, [1322](#)
- InitializeRedLUT
  - gdcm::LookupTable, [653](#)
- Initialized
  - gdcm::LookupTable, [653](#)
- Input
  - gdcm::BitmapToBitmapFilter, [210](#)
- Insert
  - gdcm::CommandDataSet, [258](#)
  - gdcm::DataSet, [332](#)
  - gdcm::FileMetaInformation, [468](#)
  - gdcm::GroupDict, [507](#)
- InsertDataElement
  - gdcm::DataSet, [332](#)
  - gdcm::Item, [604](#)
- InsertEntry
  - gdcm::Table, [1083](#)
- InstallPipeline
  - vtkImageColorViewer, [1347](#)
- Interactor
  - vtkImageColorViewer, [1355](#)
- InteractorStyle
  - vtkImageColorViewer, [1355](#)
- Internal
  - gdcm::ApplicationEntity, [123](#)
  - gdcm::Attribute, [142](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcm::Element, [396](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [411](#)
  - gdcm::LookupTable, [655](#)
  - gdcm::UI, [1131](#)
- InternalCode

- gdcm::Coder, [249](#)
- gdcm::JPEG12Codec, [609](#)
- gdcm::JPEG16Codec, [612](#)
- gdcm::JPEG8Codec, [622](#)
- Internals
  - vtkRTStructSetProperties, [1388](#)
- InverseRescale
  - gdcm::Rescaler, [904](#)
- InverseRescaleFunctionIntoBestFit
  - gdcm::Rescaler, [904](#)
- InvokeEvent
  - gdcm::Subject, [1044](#)
- IsAETitleValid
  - gdcm::network::AAAssociateRQPDU, [101](#)
- IsASCII2
  - gdcm::VR, [1265](#)
- IsASCII
  - gdcm::VR, [1265](#)
- IsBinary
  - gdcm::VR, [1265](#)
- IsBinary2
  - gdcm::VR, [1266](#)
- IsCompatible
  - gdcm::PixelFormat, [804](#)
- IsDual
  - gdcm::VR, [1266](#)
- IsEmpty
  - gdcm::Bitmap, [200](#)
  - gdcm::ByteValue, [225](#)
  - gdcm::CSAElement, [279](#)
  - gdcm::CSAHeaderDict, [292](#)
  - gdcm::Curve, [303](#)
  - gdcm::DataElement, [314](#)
  - gdcm::DataSet, [333](#)
  - gdcm::Defs, [348](#)
  - gdcm::Dict, [359](#)
  - gdcm::Dicts, [375](#)
  - gdcm::Filename, [473](#)
  - gdcm::Macros, [661](#)
  - gdcm::Modules, [702](#)
  - gdcm::Overlay, [762](#)
  - gdcm::Preamble, [831](#)
  - gdcm::PrivateDict, [858](#)
  - gdcm::SegmentHelper::BasicCodedEntry, [188](#)
  - gdcm::SequenceOfItems, [959](#)
- IsEncapsulated
  - gdcm::TransferSyntax, [1122](#)
- IsEncoded
  - gdcm::TransferSyntax, [1122](#)
- IsExplicit
  - gdcm::TransferSyntax, [1122](#)
- IsFrameEncoder
  - gdcm::ImageCodec, [547](#)
  - gdcm::JPEG2000Codec, [618](#)
  - gdcm::JPEGCodec, [629](#)
  - gdcm::JPEGLSCodec, [637](#)
  - gdcm::RLECodec, [910](#)
- IsGroupLength
  - gdcm::Tag, [1094](#)
- IsGroupXX
  - gdcm::Tag, [1095](#)
- IsIdentical
  - gdcm::Filename, [474](#)
- IsIllegal
  - gdcm::Tag, [1095](#)
- IsImage
  - gdcm::MediaStorage, [673](#)
- IsImplicit
  - gdcm::TransferSyntax, [1122](#)
- IsInPixelData
  - gdcm::Overlay, [762](#)
- IsKey
  - gdcm::Scanner, [923](#)
  - gdcm::StrictScanner, [1030](#)
- IsLastFragment
  - gdcm::network::AAAbortPDU, [88](#)
  - gdcm::network::AAAssociateACPDU, [92](#)
  - gdcm::network::AAAssociateRJPDPU, [95](#)
  - gdcm::network::AAAssociateRQPDU, [101](#)
  - gdcm::network::AReleaseRPPDU, [125](#)
  - gdcm::network::AReleaseRQPDU, [127](#)
  - gdcm::network::BasePDU, [174](#)
  - gdcm::network::PDataTFPDU, [775](#)
- IsLossless
  - gdcm::PhotometricInterpretation, [797](#)
  - gdcm::TransferSyntax, [1123](#)
- IsLossy
  - gdcm::Bitmap, [200](#)
  - gdcm::ImageCodec, [547](#)
  - gdcm::PhotometricInterpretation, [797](#)
  - gdcm::TransferSyntax, [1123](#)
- IsOdd
  - gdcm::VL, [1253](#)
- IsPresentationContextAccepted
  - gdcm::ServiceClassUser, [973](#)
- IsPrintable
  - gdcm::ByteValue, [225](#)
- IsPrivate
  - gdcm::Tag, [1095](#)
- IsPrivateCreator
  - gdcm::Tag, [1095](#)
- IsPublic
  - gdcm::Tag, [1096](#)
- IsRetired
  - gdcm::PhotometricInterpretation, [798](#)
- IsRowEncoder
  - gdcm::ImageCodec, [547](#)
  - gdcm::JPEG2000Codec, [618](#)

- gdcmm::JPEGCodec, 629
- gdcmm::JPEGLSCCodec, 637
- gdcmm::RLECodec, 911
- IsSameColorSpace
  - gdcmm::PhotometricInterpretation, 798
- IsStateSuspension
  - gdcmm::JPEG12Codec, 609
  - gdcmm::JPEG16Codec, 612
  - gdcmm::JPEG8Codec, 622
  - gdcmm::JPEGCodec, 629
- IsSwap
  - gdcmm::VR, 1266
- IsTransferSyntaxCompatible
  - gdcmm::Bitmap, 200
- IsUndefined
  - gdcmm::MediaStorage, 673
  - gdcmm::VL, 1253
- IsUndefinedLength
  - gdcmm::DataElement, 314
  - gdcmm::SequenceOfItems, 959
- IsUnique
  - gdcmm::DictEntry, 368
- IsVRFile
  - gdcmm::VR, 1266
- IsValid
  - gdcmm::ApplicationEntity, 122
  - gdcmm::BoxRegion, 214
  - gdcmm::CodeString, 253
  - gdcmm::DirectionCosines, 380
  - gdcmm::FileMetaInformation, 468
  - gdcmm::ImageCodec, 548
  - gdcmm::JPEGCodec, 629
  - gdcmm::LO, 647
  - gdcmm::PixelFormat, 804
  - gdcmm::Preamble, 831
  - gdcmm::Region, 901
  - gdcmm::String, 1036
  - gdcmm::TagPath, 1103
  - gdcmm::TransferSyntax, 1123
  - gdcmm::UIDGenerator, 1134
  - gdcmm::UUIIDGenerator, 1241
  - gdcmm::VM, 1259
  - gdcmm::VR, 1266
- IsZero
  - gdcmm::Overlay, 762
- ItFileSetHt
  - gdcmm::SerieHelper, 967
- Item
  - gdcmm::Item, 602, 603
- ItemVector
  - gdcmm::SequenceOfItems, 956
- Items
  - gdcmm::SequenceOfItems, 962
- Iterator
  - gdcmm::CSAHeaderDict, 291
  - gdcmm::DataSet, 328
  - gdcmm::Dict, 357
  - gdcmm::SequenceOfFragments, 948
  - gdcmm::SequenceOfItems, 956
- iterator
  - gdcmm::CodeString, 251
  - gdcmm::LO, 645
  - gdcmm::String, 1034
- JPEG12Codec
  - gdcmm::JPEG12Codec, 608
- JPEG16Codec
  - gdcmm::JPEG16Codec, 611
- JPEG2000Codec
  - gdcmm::JPEG2000Codec, 615
- JPEG8Codec
  - gdcmm::JPEG8Codec, 621
- JPEGCodec
  - gdcmm::JPEGCodec, 625
- JPEGLSCCodec
  - gdcmm::JPEGLSCCodec, 633
- JSON
  - gdcmm::JSON, 639
- Join
  - gdcmm::Filename, 474
- KAKADUCodec
  - gdcmm::KAKADUCodec, 642
- KeyField
  - gdcmm::CSAElement, 282
- KeyValuePairArrayType
  - gdcmm::CompositeNetworkFunctions, 262
- KeyValuePairType
  - gdcmm::CompositeNetworkFunctions, 262
- LOComp
  - gdcmm, 59
- LTComp
  - gdcmm, 59
- LUTPtr
  - gdcmm::Bitmap, 195
  - gdcmm::ImageCodec, 542
- LUT
  - gdcmm::Bitmap, 206
  - gdcmm::ImageCodec, 551
- Level
  - vtkImageMapToWindowLevelColors2, 1366
- ListCharSets
  - gdcmm::QueryFactory, 875
- LO
  - gdcmm::LO, 647
- Load
  - gdcmm::Directory, 384
  - gdcmm::MrProtocol, 711

- LoadDefault
  - gdcm::CSAHeaderDict, [292](#)
  - gdcm::Dict, [360](#)
  - gdcm::PrivateDict, [858](#)
- LoadDefaults
  - gdcm::Defs, [348](#)
  - gdcm::Dicts, [375](#)
- LoadFromDataElement
  - gdcm::CSAHeader, [288](#)
  - gdcm::PDBHeader, [781](#)
- LoadFromFile
  - gdcm::Defs, [348](#)
- LoadIconImage
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1301](#)
- LoadImageFromFiles
  - gdcm::DirectoryHelper, [388](#)
- LoadOverlays
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1301](#)
- LoadResourcesFiles
  - gdcm::Global, [504](#)
- LoadSingleFile
  - vtkGDCMImageReader, [1277](#)
  - vtkGDCMImageReader2, [1292](#)
- Locate
  - gdcm::Global, [504](#)
- LodModeType
  - gdcm, [62](#)
- LookupTable
  - gdcm::LookupTable, [650](#)
  - vtkImageMapToColors16, [1362](#)
- LookupTableType
  - gdcm::LookupTable, [650](#)
- LossyFlag
  - gdcm::Bitmap, [206](#)
  - gdcm::ImageCodec, [551](#)
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMImageReader2, [1301](#)
- m\_ConstMemberFunction
  - gdcm::MemberCommand, [679](#)
- m\_MemberFunction
  - gdcm::MemberCommand, [679](#)
  - gdcm::SimpleMemberCommand, [984](#)
- m\_This
  - gdcm::MemberCommand, [680](#)
  - gdcm::SimpleMemberCommand, [984](#)
- m\_char
  - gdcm::ignore\_char, [516](#)
- mAction
  - gdcm::network::Transition, [1128](#)
- mConnection
  - gdcm::network::ULConnectionManager, [1220](#)
- MD5DataImagesType
  - gdcm::Testing, [1105](#)
- MD5MetaImagesType
  - vtkGDCMTesting, [1326](#)
- mDataSet
  - gdcm::BaseQuery, [181](#)
- mElementOffsets
  - gdcm::StreamImageWriter, [1021](#)
- mElementOffsets1
  - gdcm::StreamImageWriter, [1021](#)
- mEnd
  - gdcm::network::Transition, [1128](#)
- mHelpDescription
  - gdcm::BaseQuery, [181](#)
  - gdcm::BaseRootQuery, [185](#)
- mImage
  - gdcm::BaseRootQuery, [186](#)
- mImplicit
  - gdcm::network::ULConnectionCallback, [1211](#)
- MPTType
  - gdcm::MeshPrimitive, [682](#)
- mPatient
  - gdcm::BaseRootQuery, [186](#)
- mRootType
  - gdcm::BaseRootQuery, [186](#)
- MSType
  - gdcm::MediaStorage, [668](#)
- mSecondaryConnection
  - gdcm::network::ULConnectionManager, [1220](#)
- mSeries
  - gdcm::BaseRootQuery, [186](#)
- mSopInstanceUID
  - gdcm::BaseQuery, [181](#)
- mStudy
  - gdcm::BaseRootQuery, [186](#)
- mTransitions
  - gdcm::network::ULConnectionManager, [1220](#)
- mWriter
  - gdcm::StreamImageWriter, [1021](#)
- mXMax
  - gdcm::StreamImageWriter, [1021](#)
- mXMin
  - gdcm::StreamImageWriter, [1022](#)
- mYMax
  - gdcm::StreamImageWriter, [1022](#)
- mYMin
  - gdcm::StreamImageWriter, [1022](#)
- mZMax
  - gdcm::StreamImageWriter, [1022](#)
- mZMin
  - gdcm::StreamImageWriter, [1022](#)
- Macro
  - gdcm::Macro, [658](#)
- MacroEntry

- gdcM, [59](#)
- Macros
  - gdcM::Macros, [661](#)
- MakeDirectory
  - gdcM::System, [1079](#)
- MakeNew
  - gdcM::network::Transition, [1127](#)
- MakeObject
  - gdcM::AnonymizeEvent, [109](#)
  - gdcM::DataEvent, [324](#)
  - gdcM::DataSetEvent, [340](#)
  - gdcM::Event, [429](#)
  - gdcM::FileNameEvent, [478](#)
  - gdcM::ProgressEvent, [865](#)
- MapCSAHeaderDictEntry
  - gdcM::CSAHeaderDict, [291](#)
- MapDictEntry
  - gdcM::Dict, [357](#)
- MapIODEntry
  - gdcM::IOD, [587](#)
- MapModuleEntry
  - gdcM::Macro, [657](#)
  - gdcM::Module, [694](#)
- MapScalarsThroughTable2
  - vtkLookupTable16, [1376](#)
- MapTableEntry
  - gdcM::Table, [1082](#)
- MappingType
  - gdcM::Scanner, [918](#)
  - gdcM::StrictScanner, [1025](#)
- MaxLength
  - gdcM::ApplicationEntity, [123](#)
  - gdcM::PersonName, [791](#)
- MaxNumberOfComponents
  - gdcM::ApplicationEntity, [123](#)
  - gdcM::PersonName, [791](#)
- MaxPrintLength
  - gdcM::Printer, [856](#)
- MaximumLengthSub
  - gdcM::network::MaximumLengthSub, [662](#)
- MediaStorage
  - gdcM::MediaStorage, [671](#)
- MediaStorageDataFilesType
  - gdcM::Testing, [1105](#)
- MedicalImageProperties
  - vtkGDCMImageReader, [1286](#)
  - vtkGDCMPolyDataReader, [1319](#)
  - vtkGDCMPolyDataWriter, [1324](#)
- MemberCommand
  - gdcM::MemberCommand, [678](#)
- MeshPrimitive
  - gdcM::MeshPrimitive, [683](#)
- MessageID
  - gdcM::network::CEchoRQ, [236](#)
- MetaInformationTS
  - gdcM::FileMetaInformation, [471](#)
- ModalityPerformedProcedureStepCreateQuery
  - gdcM::ModalityPerformedProcedureStepCreateQuery, [688](#)
- ModalityPerformedProcedureStepSetQuery
  - gdcM::ModalityPerformedProcedureStepSetQuery, [691](#)
- Mode
  - gdcM::terminal, [85](#)
- Module
  - gdcM::Module, [694](#)
- ModuleEntry
  - gdcM::ModuleEntry, [698](#)
- ModuleMapType
  - gdcM::Macros, [660](#)
  - gdcM::Modules, [701](#)
- Modules
  - gdcM::Modules, [702](#)
- MovePatientRootQuery
  - gdcM::MovePatientRootQuery, [704](#)
- MoveStudyRootQuery
  - gdcM::MoveStudyRootQuery, [707](#)
- MrProtocol
  - gdcM::MrProtocol, [710](#)
- mSPFile
  - gdcM::StreamImageWriter, [1021](#)
- NAction
  - gdcM::NormalizedNetworkFunctions, [733](#)
- NCreate
  - gdcM::NormalizedNetworkFunctions, [734](#)
- NDelete
  - gdcM::NormalizedNetworkFunctions, [734](#)
- NEventReport
  - gdcM::NormalizedNetworkFunctions, [734](#)
- NGet
  - gdcM::NormalizedNetworkFunctions, [734](#)
- NSet
  - gdcM::NormalizedNetworkFunctions, [735](#)
- Name
  - gdcM::ModuleEntry, [700](#)
- NameField
  - gdcM::CSAElement, [282](#)
  - gdcM::PDBElement, [779](#)
- NeedByteSwap
  - gdcM::Bitmap, [206](#)
  - gdcM::ImageCodec, [551](#)
- NeedOverlayCleanup
  - gdcM::ImageCodec, [551](#)
- NegotiatedType
  - gdcM::TransferSyntax, [1119](#)
- NestedMacroEntries
  - gdcM, [59](#)

## NestedModuleEntries

gdcm::NestedModuleEntries, [722](#)

## New

gdcm::Anonymizer, [115](#)  
 gdcm::FileChangeTransferSyntax, [450](#)  
 gdcm::FileStreamer, [488](#)  
 gdcm::MemberCommand, [679](#)  
 gdcm::Scanner, [923](#)  
 gdcm::SequenceOfFragments, [951](#)  
 gdcm::SequenceOfItems, [959](#)  
 gdcm::ServiceClassUser, [974](#)  
 gdcm::SimpleMemberCommand, [983](#)  
 gdcm::StrictScanner, [1030](#)  
 vtkGDCMImageReader, [1277](#)  
 vtkGDCMImageReader2, [1292](#)  
 vtkGDCMImageWriter, [1305](#)  
 vtkGDCMMedicalImageProperties, [1313](#)  
 vtkGDCMPolyDataReader, [1317](#)  
 vtkGDCMPolyDataWriter, [1322](#)  
 vtkGDCMTesting, [1328](#)  
 vtkGDCMThreadedImageReader, [1331](#)  
 vtkGDCMThreadedImageReader2, [1335](#)  
 vtkImageColorViewer, [1347](#)  
 vtkImageMapToColors16, [1359](#)  
 vtkImageMapToWindowLevelColors2, [1365](#)  
 vtkImagePlanarComponentsToComponents, [1368](#)  
 vtkImageRGBToYBR, [1371](#)  
 vtkImageYBRToRGB, [1373](#)  
 vtkLookupTable16, [1376](#)  
 vtkRTStructSetProperties, [1384](#)

## NoOfItemsField

gdcm::CSAElement, [282](#)

## Normal

gdcm::MrProtocol::Slice, [988](#)

## Normalize

gdcm::DirectionCosines, [380](#)

## NumberOfDimensions

gdcm::Bitmap, [206](#)  
 gdcm::ImageCodec, [552](#)

## NumberOfIconImages

vtkGDCMImageReader, [1287](#)  
 vtkGDCMImageReader2, [1301](#)

## NumberOfOverlays

vtkGDCMImageReader, [1287](#)  
 vtkGDCMImageReader2, [1301](#)

## NumberOfSurfaces

gdcm::SurfaceWriter, [1069](#)

## Object

gdcm::Object, [740](#)

## ObjectType

gdcm::MediaStorage, [671](#)

## Ofstream

gdcm::Writer, [1399](#)

## op

gdcm::SerieHelper::Rule, [915](#)

## OpenSSLCryptoFactory

gdcm::OpenSSLCryptoFactory, [743](#)

## OpenSSLCryptographicMessageSyntax

gdcm::OpenSSLCryptographicMessageSyntax, [745](#)

## OpenSSLP7CryptoFactory

gdcm::OpenSSLP7CryptoFactory, [748](#)

## OpenSSLP7CryptographicMessageSyntax

gdcm::OpenSSLP7CryptographicMessageSyntax,  
[750](#)

## operator const char \*

gdcm::ConstCharWrapper, [266](#)  
 gdcm::Filename, [474](#)  
 gdcm::String, [1036](#)

## operator const double \*

gdcm::DirectionCosines, [381](#)

## operator const std::vector&lt; char &gt; &amp;

gdcm::ByteValue, [226](#)

## operator MStype

gdcm::MediaStorage, [674](#)

## operator ObjectType \*

gdcm::SmartPointer, [993](#)

## operator PType

gdcm::PhotometricInterpretation, [798](#)

## operator ScalarType

gdcm::PixelFormat, [804](#)

## operator SwapCode::SwapCodeType

gdcm::SwapCode, [1072](#)

## operator TStype

gdcm::TransferSyntax, [1123](#)  
 gdcm::UIDs, [1159](#)

## operator TypeType

gdcm::Type, [1130](#)

## operator uint32\_t

gdcm::VL, [1253](#)

## operator UsageType

gdcm::Usage, [1235](#)

## operator VMType

gdcm::VM, [1259](#)

## operator VRType

gdcm::VR, [1267](#)

## operator!=

gdcm, [63](#)  
 gdcm::Attribute, [139](#)  
 gdcm::Attribute< Group, Element, TVR, VM::VM1 >,  
[146](#)  
 gdcm::CodeString, [253](#)  
 gdcm::PixelFormat, [804](#)  
 gdcm::Tag, [1096](#)

## operator&lt;

gdcm::Attribute, [139](#)  
 gdcm::Attribute< Group, Element, TVR, VM::VM1 >,  
[146](#)



- gdcmm::CSAElement, 280
- gdcmm::CSAHeaderDictEntry, 295
- gdcmm::DataElement, 314
- gdcmm::PrivateTag, 861
- gdcmm::Tag, 1096
- operator<<
  - gdcmm, 63–75
  - gdcmm::BasicOffsetTable, 191
  - gdcmm::CSAElement, 282
  - gdcmm::CSAHeader, 289
  - gdcmm::CSAHeaderDict, 292
  - gdcmm::CSAHeaderDictEntry, 296
  - gdcmm::CodeString, 254
  - gdcmm::CommandDataSet, 259
  - gdcmm::DataElement, 319
  - gdcmm::DataSet, 337
  - gdcmm::Dict, 360
  - gdcmm::DictEntry, 369
  - gdcmm::Dicts, 376
  - gdcmm::Directory, 385
  - gdcmm::File, 443
  - gdcmm::FileMetaInformation, 470
  - gdcmm::FileSet, 485
  - gdcmm::Fragment, 501
  - gdcmm::Global, 505
  - gdcmm::GroupDict, 508
  - gdcmm::IODEntry, 592
  - gdcmm::IODs, 595
  - gdcmm::IOD, 589
  - gdcmm::Item, 605
  - gdcmm::Macro, 659
  - gdcmm::Macros, 662
  - gdcmm::MediaStorage, 675
  - gdcmm::Module, 696
  - gdcmm::ModuleEntry, 700
  - gdcmm::Modules, 703
  - gdcmm::MrProtocol, 711
  - gdcmm::NestedModuleEntries, 723
  - gdcmm::Object, 741
  - gdcmm::Orientation, 755
  - gdcmm::PDBelement, 778
  - gdcmm::PDBHeader, 782
  - gdcmm::PhotometricInterpretation, 798
  - gdcmm::PixelFormat, 807
  - gdcmm::Preamble, 832
  - gdcmm::PrivateDict, 859
  - gdcmm::PrivateTag, 862
  - gdcmm::Scanner, 924
  - gdcmm::Sorter, 1002
  - gdcmm::StrictScanner, 1031
  - gdcmm::SwapCode, 1072
  - gdcmm::Table, 1083
  - gdcmm::Tag, 1100
  - gdcmm::TransferSyntax, 1123
  - gdcmm::Type, 1130
  - gdcmm::UI, 1131
  - gdcmm::Usage, 1235
  - gdcmm::Version, 1250
  - gdcmm::VL, 1255
  - gdcmm::VM, 1259
  - gdcmm::VR, 1267
- operator<=
  - gdcmm::Tag, 1096
- operator>>
  - gdcmm, 75, 76
  - gdcmm::Tag, 1100
- operator\*
  - gdcmm::SmartPointer, 993
- operator()
  - gdcmm::DataSet, 333
  - gdcmm::Scanner::ltstr, 656
  - gdcmm::StrictScanner::ltstr, 656
- operator++
  - gdcmm::VL, 1253
- operator+=
  - gdcmm::VL, 1253
- operator->
  - gdcmm::SmartPointer, 993
- operator=
  - gdcmm::BoxRegion, 214
  - gdcmm::ByteValue, 226
  - gdcmm::CSAElement, 280
  - gdcmm::DataElement, 315
  - gdcmm::DataSet, 333
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
  - gdcmm::Object, 741
  - gdcmm::Overlay, 763
  - gdcmm::ParseException, 767
  - gdcmm::Preamble, 831
  - gdcmm::SequenceOfItems, 960
  - gdcmm::SmartPointer, 993, 994
  - gdcmm::Tag, 1096
  - gdcmm::network::UserInformation, 1239
- operator==
  - gdcmm, 75
  - gdcmm::Attribute, 139
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 147
  - gdcmm::ByteValue, 226
  - gdcmm::CSAElement, 280
  - gdcmm::CodeString, 254
  - gdcmm::DataElement, 315
  - gdcmm::PDBelement, 778
  - gdcmm::PixelFormat, 804, 805
  - gdcmm::PresentationContext, 835
  - gdcmm::SequenceOfFragments, 951
  - gdcmm::SequenceOfItems, 960
  - gdcmm::Tag, 1097



- gdcmm::Value, 1246
- gdcmm::network::AbstractSyntax, 105
- gdcmm::network::PresentationContextRQ, 846
- gdcmm::network::TransferSyntaxSub, 1125
- operator[]
  - gdcmm::Attribute, 139
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 156
  - gdcmm::DataSet, 333
  - gdcmm::Element, 394
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
  - gdcmm::Tag, 1097
- OrderFileList
  - gdcmm::SerieHelper, 967
- Orientation
  - gdcmm::Orientation, 753
- OrientationType
  - gdcmm::Orientation, 753
- Output
  - gdcmm::BitmapToBitmapFilter, 210
- OutputFormat
  - vtkImageMapToColors16, 1362
- OutputTypes
  - gdcmm::DictConverter, 361
- Overlay
  - gdcmm::Overlay, 759
- OverlayImageActor
  - vtkImageColorViewer, 1355
- OverlayType
  - gdcmm::Overlay, 758
- Overlays
  - gdcmm::Pixmap, 812
- PDBElement
  - gdcmm::PDBElement, 777
- PDBHeader
  - gdcmm::PDBHeader, 780
- PDFCodec
  - gdcmm::PDFCodec, 784
- PDataTFPDU
  - gdcmm::network::PDataTFPDU, 774
- PGXCodec
  - gdcmm::PGXCodec, 793
- PIType
  - gdcmm::PhotometricInterpretation, 796
- PNComp
  - gdcmm, 59
- PNMCodec
  - gdcmm::PNMCodec, 826
- PVRGCodec
  - gdcmm::PVRGCodec, 867
- Pack
  - gdcmm::Unpacker12Bits, 1232
- Padding
  - gdcmm::ApplicationEntity, 123
  - gdcmm::PersonName, 791
- Parent
  - gdcmm::Element< TVR, VM::VM1\_2 >, 397
  - gdcmm::Element< TVR, VM::VM2\_2n >, 404
  - gdcmm::Element< TVR, VM::VM2\_n >, 406
  - gdcmm::Element< TVR, VM::VM3\_3n >, 408
  - gdcmm::Element< TVR, VM::VM3\_n >, 410
- Parse
  - gdcmm::Parser, 771
- ParseBuffer
  - gdcmm::Parser, 771
- ParseCertificateFile
  - gdcmm::CAPICryptographicMessageSyntax, 233
  - gdcmm::CryptographicMessageSyntax, 274
  - gdcmm::OpenSSLCryptographicMessageSyntax, 746
  - gdcmm::OpenSSLP7CryptographicMessageSyntax, 751
- ParseDateTime
  - gdcmm::System, 1079
- ParseDump
  - gdcmm::ASN1, 131
- ParseDumpFile
  - gdcmm::ASN1, 131
- ParseException
  - gdcmm::ParseException, 767
- ParseKeyFile
  - gdcmm::CAPICryptographicMessageSyntax, 233
  - gdcmm::CryptographicMessageSyntax, 274
  - gdcmm::OpenSSLCryptographicMessageSyntax, 746
  - gdcmm::OpenSSLP7CryptographicMessageSyntax, 751
- Parser
  - gdcmm::Parser, 770
- PassAlphaToOutput
  - vtkImageMapToColors16, 1363
- Patient
  - gdcmm::Patient, 772
- PerformAction
  - gdcmm::network::ULAction, 1163
  - gdcmm::network::ULActionAA1, 1164
  - gdcmm::network::ULActionAA2, 1165
  - gdcmm::network::ULActionAA3, 1167
  - gdcmm::network::ULActionAA4, 1168
  - gdcmm::network::ULActionAA5, 1169
  - gdcmm::network::ULActionAA6, 1171
  - gdcmm::network::ULActionAA7, 1172
  - gdcmm::network::ULActionAA8, 1173
  - gdcmm::network::ULActionAE1, 1175
  - gdcmm::network::ULActionAE2, 1176
  - gdcmm::network::ULActionAE3, 1177
  - gdcmm::network::ULActionAE4, 1179
  - gdcmm::network::ULActionAE5, 1180
  - gdcmm::network::ULActionAE6, 1181

- gdcm::network::ULActionAE7, [1183](#)
- gdcm::network::ULActionAE8, [1184](#)
- gdcm::network::ULActionAR1, [1185](#)
- gdcm::network::ULActionAR10, [1187](#)
- gdcm::network::ULActionAR2, [1188](#)
- gdcm::network::ULActionAR3, [1189](#)
- gdcm::network::ULActionAR4, [1191](#)
- gdcm::network::ULActionAR5, [1192](#)
- gdcm::network::ULActionAR6, [1193](#)
- gdcm::network::ULActionAR7, [1195](#)
- gdcm::network::ULActionAR8, [1196](#)
- gdcm::network::ULActionAR9, [1197](#)
- gdcm::network::ULActionDT1, [1199](#)
- gdcm::network::ULActionDT2, [1200](#)
- PF
  - gdcm::Bitmap, [206](#)
  - gdcm::ImageCodec, [552](#)
- PhotometricInterpretation
  - gdcm::PhotometricInterpretation, [796](#)
- PI
  - gdcm::Bitmap, [206](#)
  - gdcm::ImageCodec, [552](#)
- PixelData
  - gdcm::Bitmap, [207](#)
  - gdcm::PixmapReader, [817](#)
  - gdcm::PixmapWriter, [824](#)
- PixelFormat
  - gdcm::PixelFormat, [801](#)
- Pixmap
  - gdcm::Pixmap, [809](#)
- PixmapReader
  - gdcm::Bitmap, [205](#)
  - gdcm::PixmapReader, [815](#)
- PixmapToPixmapFilter
  - gdcm::PixmapToPixmapFilter, [819](#)
- PixmapWriter
  - gdcm::PixmapWriter, [822](#)
- PlanarConfiguration
  - gdcm::Bitmap, [207](#)
  - gdcm::ImageCodec, [552](#)
  - vtkGDCMImageReader, [1287](#)
  - vtkGDCMImageReader2, [1301](#)
- pointer
  - gdcm::CodeString, [251](#)
  - gdcm::LO, [646](#)
  - gdcm::String, [1034](#)
- Position
  - gdcm::MrProtocol::Slice, [988](#)
- Preamble
  - gdcm::Preamble, [829](#), [830](#)
- PrepareWrite
  - gdcm::PixmapWriter, [823](#)
  - gdcm::SegmentWriter, [944](#)
  - gdcm::SurfaceWriter, [1069](#)
- PrepareWritePointMacro
  - gdcm::SurfaceWriter, [1069](#)
- Prepend
  - gdcm::Global, [505](#)
- PresentationContext
  - gdcm::PresentationContext, [834](#)
- PresentationContextAC
  - gdcm::network::PresentationContextAC, [837](#)
- PresentationContextArrayType
  - gdcm::PresentationContextGenerator, [841](#)
  - gdcm::network::AAAssociateRQPDU, [98](#)
- PresentationContextGenerator
  - gdcm::PresentationContextGenerator, [841](#)
- PresentationContextRQ
  - gdcm::network::PresentationContextRQ, [844](#)
- PresentationDataValue
  - gdcm::network::PresentationDataValue, [848](#)
- PrettyPrintOff
  - gdcm::JSON, [640](#)
- PrettyPrintOn
  - gdcm::JSON, [640](#)
- PrimitiveData
  - gdcm::MeshPrimitive, [685](#)
- PrimitiveType
  - gdcm::MeshPrimitive, [686](#)
- PrimitivesData
  - gdcm::MeshPrimitive, [682](#)
- Print
  - gdcm::ApplicationEntity, [122](#)
  - gdcm::Attribute, [140](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [147](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [156](#)
  - gdcm::BaseQuery, [179](#)
  - gdcm::Bitmap, [200](#)
  - gdcm::BoxRegion, [215](#)
  - gdcm::ByteValue, [226](#)
  - gdcm::CSAHeader, [288](#)
  - gdcm::Curve, [304](#)
  - gdcm::DataSet, [333](#)
  - gdcm::DictPrinter, [371](#)
  - gdcm::DirectionCosines, [381](#)
  - gdcm::Directory, [385](#)
  - gdcm::Element, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [401](#)
  - gdcm::Element< VR::AS, VM::VM5 >, [411](#)
  - gdcm::Event, [429](#)
  - gdcm::Image, [520](#)
  - gdcm::LookupTable, [653](#)
  - gdcm::MrProtocol, [711](#)
  - gdcm::Object, [741](#)
  - gdcm::Orientation, [754](#)
  - gdcm::Overlay, [763](#)

- gdcm::PDBHeader, [782](#)
- gdcm::PersonName, [790](#)
- gdcm::PixelFormat, [805](#)
- gdcm::Pixmap, [811](#)
- gdcm::Preamble, [831](#)
- gdcm::PresentationContext, [836](#)
- gdcm::Printer, [854](#)
- gdcm::Region, [901](#)
- gdcm::Scanner, [923](#)
- gdcm::SegmentedPaletteColorLookupTable, [937](#)
- gdcm::SequenceOfFragments, [951](#)
- gdcm::SequenceOfItems, [960](#)
- gdcm::Sorter, [1001](#)
- gdcm::StrictScanner, [1030](#)
- gdcm::TagPath, [1103](#)
- gdcm::Testing, [1111](#)
- gdcm::Version, [1250](#)
- gdcm::XMLPrinter, [1405](#)
- gdcm::network::AAAbortPDU, [88](#)
- gdcm::network::AAssociateACPDU, [93](#)
- gdcm::network::AAssociateRJPDU, [96](#)
- gdcm::network::AAssociateRQPDU, [101](#)
- gdcm::network::AReleaseRPPDU, [125](#)
- gdcm::network::AReleaseRQPDU, [128](#)
- gdcm::network::AbstractSyntax, [105](#)
- gdcm::network::ApplicationContext, [120](#)
- gdcm::network::AsynchronousOperationsWindow←  
Sub, [133](#)
- gdcm::network::BasePDU, [175](#)
- gdcm::network::ImplementationClassUIDSub, [579](#)
- gdcm::network::ImplementationVersionNameSub,  
[582](#)
- gdcm::network::MaximumLengthSub, [663](#)
- gdcm::network::PDataTFPDU, [775](#)
- gdcm::network::PresentationContextAC, [838](#)
- gdcm::network::PresentationContextRQ, [846](#)
- gdcm::network::PresentationDataValue, [849](#)
- gdcm::network::RoleSelectionSub, [913](#)
- gdcm::network::SOPClassExtendedNegociationSub,  
[995](#)
- gdcm::network::ServiceClassApplicationInformation,  
[969](#)
- gdcm::network::TransferSyntaxSub, [1125](#)
- gdcm::network::UserInfo, [1239](#)
- PrintASCIIXML
  - gdcm::ByteValue, [227](#)
- PrintASCII
  - gdcm::ByteValue, [226](#)
- PrintAsContinuousString
  - gdcm::Tag, [1097](#)
- PrintAsContinuousUpperCaseString
  - gdcm::Tag, [1097](#)
- PrintAsPipeSeparatedString
  - gdcm::Tag, [1098](#)
- PrintDataElement
  - gdcm::Printer, [854](#)
  - gdcm::XMLPrinter, [1405](#)
- PrintDataElement2
  - gdcm::DictPrinter, [371](#)
- PrintDataSet
  - gdcm::Printer, [854](#)
  - gdcm::XMLPrinter, [1405](#)
- PrintDataSet2
  - gdcm::DictPrinter, [372](#)
- PrintGroupLength
  - gdcm::ByteValue, [227](#)
- PrintHex
  - gdcm::ByteValue, [227](#)
- PrintHexXML
  - gdcm::ByteValue, [227](#)
- PrintPNXML
  - gdcm::ByteValue, [227](#)
- PrintSelf
  - vtkGDCMImageReader, [1277](#)
  - vtkGDCMImageReader2, [1292](#)
  - vtkGDCMImageWriter, [1306](#)
  - vtkGDCMMedicalImageProperties, [1314](#)
  - vtkGDCMPolyDataReader, [1317](#)
  - vtkGDCMPolyDataWriter, [1323](#)
  - vtkGDCMTesting, [1328](#)
  - vtkGDCMThreadedImageReader, [1331](#)
  - vtkGDCMThreadedImageReader2, [1335](#)
  - vtkImageColorViewer, [1348](#)
  - vtkImageMapToColors16, [1359](#)
  - vtkImageMapToWindowLevelColors2, [1365](#)
  - vtkImagePlanarComponentsToComponents, [1368](#)
  - vtkImageRGBToYBR, [1371](#)
  - vtkImageYBRToRGB, [1373](#)
  - vtkLookupTable16, [1377](#)
  - vtkRTStructSetProperties, [1384](#)
- PrintSQ
  - gdcm::Printer, [855](#)
  - gdcm::XMLPrinter, [1405](#)
- PrintStyle
  - gdcm::Printer, [856](#)
  - gdcm::XMLPrinter, [1406](#)
- PrintStyles
  - gdcm::Printer, [853](#)
  - gdcm::XMLPrinter, [1404](#)
- PrintTable
  - gdcm::network::ULTransitionTable, [1224](#)
- PrintXML
  - gdcm::PrivateDict, [858](#)
- Printer
  - gdcm::Printer, [853](#)
- PrivateDict
  - gdcm::PrivateDict, [857](#)
- PrivateTag

- gdcmm::PrivateTag, 860, 861
- Process
  - gdcmm::Parser, 771
- ProcessDataSet
  - gdcmm::FileExplicitFilter, 460
- ProcessPublicTag
  - gdcmm::Scanner, 923
  - gdcmm::StrictScanner, 1030
- ProcessRequest
  - vtkGDCMImageReader2, 1292
- ProduceCharacterSetDataElement
  - gdcmm::QueryFactory, 875
- ProduceQuery
  - gdcmm::QueryFactory, 875
- ProgressEvent
  - gdcmm::ProgressEvent, 864
- PropertyCategory
  - gdcmm::Segment, 934
- PropertyType
  - gdcmm::Segment, 934
- PropertyTypeModifiers
  - gdcmm::Segment, 934
- Push
  - gdcmm::TagPath, 1103
- PushBackFile
  - vtkGDCMMedicalImageProperties, 1314
- PythonFilter
  - gdcmm::PythonFilter, 870
- Quality
  - gdcmm::JPEGCodec, 631
- QueryFactory
  - gdcmm::BaseQuery, 181
  - gdcmm::BaseRootQuery, 185
  - gdcmm::FindPatientRootQuery, 495
  - gdcmm::FindStudyRootQuery, 498
  - gdcmm::ModalityPerformedProcedureStepCreateQuery, 688
  - gdcmm::ModalityPerformedProcedureStepSetQuery, 691
  - gdcmm::MovePatientRootQuery, 706
  - gdcmm::MoveStudyRootQuery, 709
  - gdcmm::WLMFindQuery, 1393
- RAWCodec
  - gdcmm::RAWCodec, 887
- README.txt, 1684
- RGB2YBR
  - gdcmm::ImageChangePhotometricInterpretation, 529
- RGBPixelsToRGBPlanes
  - gdcmm::ImageChangePlanarConfiguration, 533
- RGBPlanesToRGBPixels
  - gdcmm::ImageChangePlanarConfiguration, 533
- RGBToRecommendedDisplayCIELab
  - gdcmm::SurfaceHelper, 1062
- RGBToRecommendedDisplayGrayscale
  - gdcmm::SurfaceHelper, 1063
- RLECodec
  - gdcmm::RLECodec, 908
- RTStructSetProperties
  - vtkGDCMPolyDataReader, 1320
  - vtkGDCMPolyDataWriter, 1325
- Read
  - gdcmm::BasicOffsetTable, 191
  - gdcmm::ByteValue, 227, 228
  - gdcmm::CP246ExplicitDataElement, 268
  - gdcmm::CSAHeader, 289
  - gdcmm::CommandDataSet, 258
  - gdcmm::DataElement, 315
  - gdcmm::DataSet, 334
  - gdcmm::Element, 395
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
  - gdcmm::EncodingImplementation< VR::VRASCII >, 421
  - gdcmm::EncodingImplementation< VR::VRBINARY >, 423
  - gdcmm::ExplicitDataElement, 435
  - gdcmm::ExplicitImplicitDataElement, 438
  - gdcmm::File, 443
  - gdcmm::FileMetaInformation, 468
  - gdcmm::Fragment, 500
  - gdcmm::ImageReader, 567
  - gdcmm::ImageRegionReader, 571
  - gdcmm::ImplicitDataElement, 584
  - gdcmm::Item, 604
  - gdcmm::PGXCodec, 794
  - gdcmm::PNMCodec, 827
  - gdcmm::PixmapReader, 816
  - gdcmm::Preamble, 831
  - gdcmm::Reader, 894
  - gdcmm::SegmentReader, 940
  - gdcmm::SequenceOfFragments, 952
  - gdcmm::SequenceOfItems, 960
  - gdcmm::StreamImageReader, 1013
  - gdcmm::SurfaceReader, 1066
  - gdcmm::TableReader, 1088
  - gdcmm::Tag, 1098
  - gdcmm::UNExplicitDataElement, 1229
  - gdcmm::UNExplicitImplicitDataElement, 1231
  - gdcmm::VR16ExplicitDataElement, 1269
  - gdcmm::VRVLSize< 0 >, 1271
  - gdcmm::VRVLSize< 1 >, 1272
  - gdcmm::ValueIO, 1247
  - gdcmm::VL, 1254
  - gdcmm::VR, 1267
  - gdcmm::network::AAbortPDU, 88
  - gdcmm::network::AAssociateACPDU, 93
  - gdcmm::network::AAssociateRJPDU, 96
  - gdcmm::network::AAssociateRQPDU, 101

- gdcm::network::AReleaseRPPDU, 125
- gdcm::network::AReleaseRQPDU, 128
- gdcm::network::AbstractSyntax, 106
- gdcm::network::ApplicationContext, 120
- gdcm::network::AsynchronousOperationsWindow↔  
Sub, 133
- gdcm::network::BasePDU, 175
- gdcm::network::ImplementationClassUIDSub, 579
- gdcm::network::ImplementationVersionNameSub,  
582
- gdcm::network::MaximumLengthSub, 663
- gdcm::network::PDataTFPDU, 775
- gdcm::network::PresentationContextAC, 838
- gdcm::network::PresentationContextRQ, 846
- gdcm::network::PresentationDataValue, 849
- gdcm::network::RoleSelectionSub, 913
- gdcm::network::SOPClassExtendedNegociationSub,  
995
- gdcm::network::ServiceClassApplicationInformation,  
969
- gdcm::network::TransferSyntaxSub, 1125
- gdcm::network::UserInformation, 1239
- Read16
  - gdcm::VL, 1254
- ReadACRNEMAIImage
  - gdcm::ImageReader, 567
  - gdcm::PixmapReader, 816
- ReadBacktrack
  - gdcm::Fragment, 500
- ReadCompat
  - gdcm::FileMetaInformation, 468
- ReadCompatInternal
  - gdcm::FileMetaInformation, 469
- ReadComputeLength
  - gdcm::EncodingImplementation< VR::VRASCII >,  
421
  - gdcm::EncodingImplementation< VR::VRBINARY  
>, 423
- ReadDataSet
  - gdcm::Reader, 894
- ReadFiles
  - vtkGDCMThreadedImageReader, 1331
- ReadFromCommaSeparatedString
  - gdcm::PrivateTag, 861
  - gdcm::Tag, 1098
- ReadFromContinuousString
  - gdcm::Tag, 1098
- ReadFromPipeSeparatedString
  - gdcm::Tag, 1098
- ReadImage
  - gdcm::ImageReader, 567
  - gdcm::PixmapReader, 816
- ReadImageInformation
  - gdcm::StreamImageReader, 1014
- ReadImageInternal
  - gdcm::PixmapReader, 816
- ReadInformation
  - gdcm::ImageRegionReader, 571
- ReadInto
  - gdcm::network::PDataTFPDU, 775
  - gdcm::network::PresentationDataValue, 849
- ReadIntoBuffer
  - gdcm::ImageRegionReader, 571
- ReadMetaInformation
  - gdcm::Reader, 895
- ReadNested
  - gdcm::DataSet, 334
- ReadNoSwap
  - gdcm::EncodingImplementation< VR::VRASCII >,  
421
  - gdcm::EncodingImplementation< VR::VRBINARY  
>, 423
- ReadOrSkip
  - gdcm::DataElement, 315
- ReadPointMacro
  - gdcm::SurfaceReader, 1066
- ReadPreValue
  - gdcm::CP246ExplicitDataElement, 268
  - gdcm::DataElement, 315
  - gdcm::ExplicitDataElement, 435
  - gdcm::ExplicitImplicitDataElement, 438
  - gdcm::Fragment, 500
  - gdcm::ImplicitDataElement, 584
  - gdcm::SequenceOfFragments, 952
  - gdcm::UNExplicitDataElement, 1229
  - gdcm::UNExplicitImplicitDataElement, 1231
  - gdcm::VR16ExplicitDataElement, 1269
- ReadPreamble
  - gdcm::Reader, 895
- ReadSegment
  - gdcm::SegmentReader, 941
- ReadSegments
  - gdcm::SegmentReader, 941
- ReadSelectedPrivateTags
  - gdcm::DataSet, 334
  - gdcm::Reader, 895
- ReadSelectedPrivateTagsWithLength
  - gdcm::DataSet, 334
- ReadSelectedTags
  - gdcm::DataSet, 334
  - gdcm::Reader, 895
- ReadSelectedTagsWithLength
  - gdcm::DataSet, 335
- ReadSurface
  - gdcm::SurfaceReader, 1066
- ReadSurfaces
  - gdcm::SurfaceReader, 1066
- ReadUpToTag

- gdcmm::DataSet, 335
- gdcmm::Reader, 895
- ReadUpToTagWithLength
  - gdcmm::DataSet, 335
- ReadValue
  - gdcmm::CP246ExplicitDataElement, 268
  - gdcmm::DataElement, 316
  - gdcmm::ExplicitDataElement, 435
  - gdcmm::ExplicitImplicitDataElement, 438
  - gdcmm::Fragment, 501
  - gdcmm::ImplicitDataElement, 585
  - gdcmm::SequenceOfFragments, 952
  - gdcmm::UNExplicitDataElement, 1229
  - gdcmm::UNExplicitImplicitDataElement, 1231
  - gdcmm::VR16ExplicitDataElement, 1269
- ReadValueWithLength
  - gdcmm::DataElement, 316
  - gdcmm::ImplicitDataElement, 585
- ReadVM
  - gdcmm::DictConverter, 363
- ReadVR
  - gdcmm::DictConverter, 364
- ReadWithLength
  - gdcmm::CP246ExplicitDataElement, 269
  - gdcmm::DataElement, 316
  - gdcmm::DataSet, 335
  - gdcmm::ExplicitDataElement, 435
  - gdcmm::ExplicitImplicitDataElement, 438
  - gdcmm::ImplicitDataElement, 585
  - gdcmm::UNExplicitDataElement, 1229
  - gdcmm::VR16ExplicitDataElement, 1270
- Reader
  - gdcmm::Reader, 893
- Readuint16
  - gdcmm::DictConverter, 363
- RealWorldValueIntercept
  - gdcmm::RealWorldValueMappingContent, 898
- RealWorldValueSlope
  - gdcmm::RealWorldValueMappingContent, 898
- RecommendedDisplayCIELabToRGB
  - gdcmm::SurfaceHelper, 1061
- RecurseDataSet
  - gdcmm::Anonymizer, 115
- reference
  - gdcmm::CodeString, 251
  - gdcmm::LO, 646
  - gdcmm::String, 1035
- ReferenceFrameOfReferenceUID
  - vtkRTStructSetProperties, 1388
- ReferenceSeriesInstanceUID
  - vtkRTStructSetProperties, 1388
- Region
  - gdcmm::Region, 899
- Register
  - gdcmm::Object, 741
- Remove
  - gdcmm::Anonymizer, 115
  - gdcmm::DataSet, 335
  - gdcmm::FileAnonymizer, 446
  - gdcmm::Preamble, 831
- RemoveAllObservers
  - gdcmm::Subject, 1045
- RemoveDictEntry
  - gdcmm::PrivateDict, 858
- RemoveFile
  - gdcmm::System, 1080
- RemoveGroupLength
  - gdcmm::Anonymizer, 115
- RemoveItemByIndex
  - gdcmm::SequenceOfItems, 961
- RemoveObserver
  - gdcmm::Subject, 1045
- RemoveOverlay
  - gdcmm::Pixmap, 811
- RemovePrivateTags
  - gdcmm::Anonymizer, 116
- RemoveRetired
  - gdcmm::Anonymizer, 116
- Render
  - vtkImageColorViewer, 1348
- RenderWindow
  - vtkImageColorViewer, 1356
- Renderer
  - vtkImageColorViewer, 1355
- Replace
  - gdcmm::Anonymizer, 116
  - gdcmm::CommandDataSet, 259
  - gdcmm::DataSet, 336
  - gdcmm::FileAnonymizer, 447
  - gdcmm::FileMetaInformation, 469
- ReplaceEmpty
  - gdcmm::DataSet, 336
- RequestData
  - vtkGDCMImageReader2, 1293
  - vtkGDCMPolyDataReader, 1317
  - vtkImageMapToColors16, 1359
  - vtkImageMapToWindowLevelColors2, 1365
  - vtkImagePlanarComponentsToComponents, 1369
- RequestData\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, 1318
- RequestData\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, 1318
- RequestDataCompat
  - vtkGDCMImageReader, 1278
  - vtkGDCMImageReader2, 1293
  - vtkGDCMThreadedImageReader, 1332
- RequestInformation
  - vtkGDCMImageReader2, 1293



- vtkGDCMPolyDataReader, [1318](#)
  - vtkGDCMThreadedImageReader2, [1336](#)
  - vtkImageMapToColors16, [1359](#)
  - vtkImageMapToWindowLevelColors2, [1365](#)
- RequestInformation\_HemodynamicWaveformStorage
  - vtkGDCMPolyDataReader, [1318](#)
- RequestInformation\_RTStructureSetStorage
  - vtkGDCMPolyDataReader, [1318](#)
- RequestInformationCompat
  - vtkGDCMImageReader, [1278](#)
  - vtkGDCMImageReader2, [1293](#)
- RequestPaddedCompositePixelCode
  - gdcm::ImageCodec, [552](#)
- RequestPlanarConfiguration
  - gdcm::ImageCodec, [552](#)
- Rescale
  - gdcm::Rescaler, [904](#)
- RescaleFunctionIntoBestFit
  - gdcm::Rescaler, [904](#)
- Rescaler
  - gdcm::Rescaler, [903](#)
- ReserveDataElement
  - gdcm::FileStreamer, [488](#)
- ReserveGroupDataElement
  - gdcm::FileStreamer, [488](#)
- ResetHandledDataSet
  - gdcm::network::ULConnectionCallback, [1210](#)
- RetrieveSOPInstanceUIDFromIndex
  - gdcm::DirectoryHelper, [388](#)
- RetrieveSOPInstanceUIDFromZPosition
  - gdcm::DirectoryHelper, [388](#)
- reverse\_iterator
  - gdcm::CodeString, [251](#)
  - gdcm::LO, [646](#)
  - gdcm::String, [1035](#)
- RoleSelectionSub
  - gdcm::network::RoleSelectionSub, [912](#)
- Round
  - gdcm, [76](#)
- RunEventLoop
  - gdcm::network::ULConnectionManager, [1216](#)
- RunMoveEventLoop
  - gdcm::network::ULConnectionManager, [1217](#)
- SHA1
  - gdcm::SHA1, [979](#)
- SHComp
  - gdcm, [59](#)
- SOPClassExtendedNegociationSub
  - gdcm::network::SOPClassExtendedNegociationSub, [995](#)
- SOPClassUIDMode
  - gdcm::EmptyMaskGenerator, [418](#)
- SOPInstanceUID
  - vtkRTStructSetProperties, [1388](#)
- STATES
  - gdcm::Surface, [1048](#)
- STComp
  - gdcm, [60](#)
- ScalarType
  - gdcm::PixelFormat, [800](#)
- Scale
  - vtkGDCMImageReader, [1287](#)
  - vtkGDCMImageReader2, [1302](#)
- Scan
  - gdcm::Scanner, [924](#)
  - gdcm::StrictScanner, [1031](#)
- Scanner
  - gdcm::Scanner, [919](#)
- Segment
  - gdcm::Segment, [928](#)
- SegmentAlgorithmName
  - gdcm::Segment, [934](#)
- SegmentAlgorithmType
  - gdcm::Segment, [934](#)
- SegmentDescription
  - gdcm::Segment, [934](#)
- SegmentLabel
  - gdcm::Segment, [934](#)
- SegmentMap
  - gdcm::SegmentReader, [939](#)
- SegmentNumber
  - gdcm::Segment, [935](#)
- SegmentReader
  - gdcm::SegmentReader, [940](#)
- SegmentVector
  - gdcm::SegmentReader, [940](#)
  - gdcm::SegmentWriter, [943](#)
- SegmentWriter
  - gdcm::SegmentWriter, [943](#)
- SegmentedPaletteColorLookupTable
  - gdcm::SegmentedPaletteColorLookupTable, [936](#)
- Segments
  - gdcm::SegmentReader, [941](#)
  - gdcm::SegmentWriter, [945](#)
- Selection
  - gdcm::Sorter, [1002](#)
- SelectionMap
  - gdcm::Sorter, [1000](#)
- Self
  - gdcm::AnonymizeEvent, [108](#)
  - gdcm::DataSetEvent, [323](#)
  - gdcm::DataSetEvent, [339](#)
  - gdcm::FileNameEvent, [477](#)
  - gdcm::MemberCommand, [677](#)
  - gdcm::ProgressEvent, [864](#)
  - gdcm::SimpleMemberCommand, [982](#)
- SendEcho

- gdcmm::ServiceClassUser, 974
- gdcmm::network::ULConnectionManager, 1217
- SendFind
  - gdcmm::ServiceClassUser, 974
  - gdcmm::network::ULConnectionManager, 1217
- SendMove
  - gdcmm::ServiceClassUser, 974, 975
  - gdcmm::network::ULConnectionManager, 1217
- SendNAAction
  - gdcmm::network::ULConnectionManager, 1218
- SendNCreate
  - gdcmm::network::ULConnectionManager, 1218
- SendNDelete
  - gdcmm::network::ULConnectionManager, 1218
- SendNEventReport
  - gdcmm::network::ULConnectionManager, 1219
- SendNGet
  - gdcmm::network::ULConnectionManager, 1219
- SendNSet
  - gdcmm::network::ULConnectionManager, 1219
- SendStore
  - gdcmm::ServiceClassUser, 975
  - gdcmm::network::ULConnectionManager, 1220
- Separator
  - gdcmm::ApplicationEntity, 124
  - gdcmm::PersonName, 791
- SequenceLengthField
  - gdcmm::SequenceOfItems, 962
- SequenceOfFragments
  - gdcmm::SequenceOfFragments, 948
- SequenceOfItems
  - gdcmm::SequenceOfItems, 956
- SerieHelper
  - gdcmm::SerieHelper, 964
- SerieRestrictions
  - gdcmm::SerieHelper, 964
- Series
  - gdcmm::Series, 968
- SeriesInstanceUID
  - vtkRTStructSetProperties, 1388
- ServiceClassApplicationInformation
  - gdcmm::network::ServiceClassApplicationInformation, 969
- ServiceClassUser
  - gdcmm::ServiceClassUser, 972
- Set
  - gdcmm::Attribute, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 147
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 156
  - gdcmm::Element, 395
  - gdcmm::Element< TVR, VM::VM1\_n >, 401
- SetAETitle
  - gdcmm::ServiceClassUser, 975
- SetAbstractSyntax
  - gdcmm::PresentationContext, 836
  - gdcmm::network::PresentationContextRQ, 846
- SetAlgorithmFamily
  - gdcmm::Surface, 1055
- SetAlgorithmName
  - gdcmm::Surface, 1055
- SetAlgorithmVersion
  - gdcmm::Surface, 1055
- SetAnatomicRegion
  - gdcmm::Segment, 931
- SetAnatomicRegionModifiers
  - gdcmm::Segment, 931
- SetAppendDerivationHistory
  - gdcmm::FileDerivation, 457
- SetArray
  - gdcmm::Element< TVR, VM::VM1\_n >, 402
- SetAxisOfRotation
  - gdcmm::Surface, 1055
- SetBitPosition
  - gdcmm::Overlay, 763
- SetBitSample
  - gdcmm::JPEGCodec, 629
- SetBitsAllocated
  - gdcmm::Overlay, 763
  - gdcmm::PixelFormat, 805
- SetBitsStored
  - gdcmm::PixelFormat, 805
- SetBlob
  - gdcmm::ApplicationEntity, 123
  - gdcmm::PersonName, 790
  - gdcmm::network::PresentationDataValue, 850
- SetBlueLUT
  - gdcmm::LookupTable, 654
- SetBufferLength
  - gdcmm::JPEGLSCCodec, 637
  - gdcmm::PNMCodec, 828
  - gdcmm::RLECodec, 911
- SetByteSwapTag
  - gdcmm::ByteSwapFilter, 220
- SetByteValue
  - gdcmm::Attribute, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 147
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >, 156
  - gdcmm::CSAElement, 280
  - gdcmm::DataElement, 316
- SetByteValueNoSwap
  - gdcmm::Attribute, 140
  - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 147
- SetCallbackFunction



- gdcm::MemberCommand, 679
- gdcm::SimpleMemberCommand, 983
- SetCalledAETitle
  - gdcm::ServiceClassUser, 976
  - gdcm::network::AAAssociateACPDU, 93
  - gdcm::network::AAAssociateRQPDU, 102
- SetCallingAETitle
  - gdcm::network::AAAssociateACPDU, 93
  - gdcm::network::AAAssociateRQPDU, 102
- SetCenterOfRotation
  - gdcm::Surface, 1055
- SetChangePrivateTags
  - gdcm::FileExplicitFilter, 461
- SetCheckFileMetaInformation
  - gdcm::Writer, 1397
- SetCipherType
  - gdcm::CAPICryptographicMessageSyntax, 233
  - gdcm::CryptographicMessageSyntax, 274
  - gdcm::OpenSSLCryptographicMessageSyntax, 746
  - gdcm::OpenSSLP7CryptographicMessageSyntax, 751
- SetColor
  - gdcm::Printer, 855
- SetColorLevel
  - vtkImageColorViewer, 1348
- SetColorWindow
  - vtkImageColorViewer, 1348
- SetColumns
  - gdcm::Bitmap, 200
  - gdcm::Overlay, 763
- SetCommand
  - gdcm::network::PresentationDataValue, 850
- SetComponents
  - gdcm::PersonName, 790
- SetCompressIconImage
  - gdcm::ImageChangeTransferSyntax, 537
- SetComputeZSpacing
  - gdcm::IPPSorter, 598
- SetCoordinateStartValue
  - gdcm::Curve, 304
- SetCoordinateStepValue
  - gdcm::Curve, 304
- SetCryptographicMessageSyntax
  - gdcm::Anonymizer, 117
- SetCurve
  - gdcm::Curve, 304
  - vtkGDCMImageReader, 1278
  - vtkGDCMImageReader2, 1293
- SetCurveDataDescriptor
  - gdcm::Curve, 304
- SetCurveDescription
  - gdcm::Curve, 304
- SetData
  - gdcm::DataEvent, 325
- SetDataElement
  - gdcm::Bitmap, 201
- SetDataSet
  - gdcm::File, 443
  - gdcm::network::PresentationDataValue, 850
- SetDataSetTransferSyntax
  - gdcm::FileMetaInformation, 469
- SetDataValueRepresentation
  - gdcm::Curve, 305
- SetDebug
  - gdcm::Trace, 1115
- SetDebugStream
  - gdcm::Trace, 1115
- SetDefaultTransferSyntax
  - gdcm::PresentationContextGenerator, 842
- SetDerivationCodeSequenceCodeValue
  - gdcm::FileDerivation, 457
- SetDerivationDescription
  - gdcm::FileDerivation, 458
- SetDescription
  - gdcm::CSAHeaderDictEntry, 295
  - gdcm::ModuleEntry, 699
  - gdcm::Overlay, 764
- SetDescriptor
  - gdcm::DICOMDIRGenerator, 355
- SetDictName
  - gdcm::DictConverter, 364
- SetDicts
  - gdcm::PythonFilter, 870
  - gdcm::StringFilter, 1039
- SetDimension
  - gdcm::Bitmap, 201
- SetDimensions
  - gdcm::Bitmap, 201
  - gdcm::Curve, 305
  - gdcm::ImageCodec, 548
- SetDimensionsValue
  - gdcm::ImageHelper, 562
- SetDirectionCosines
  - gdcm::Image, 520, 521
  - vtkGDCMImageWriter, 1306
- SetDirectionCosinesFromImageOrientationPatient
  - vtkGDCMImageWriter, 1306
- SetDirectionCosinesTolerance
  - gdcm::IPPSorter, 598
- SetDirectionCosinesValue
  - gdcm::ImageHelper, 562
- SetDirectory
  - gdcm::SerieHelper, 967
  - gdcm::network::ULWritingCallback, 1226
- SetDisplayId
  - vtkImageColorViewer, 1348
- SetDomain
  - gdcm::BoxRegion, 215

- SetDropDuplicatePositions
  - gdcm::IPPSorter, 599
- SetElement
  - gdcm::Tag, 1099
- SetElementHandler
  - gdcm::Parser, 771
- SetElementTag
  - gdcm::Tag, 1099
- SetElementXX
  - gdcm::DictEntry, 368
- SetError
  - gdcm::Trace, 1115
- SetErrorStream
  - gdcm::Trace, 1116
- SetEvent
  - gdcm::network::ULEvent, 1223
- SetFile
  - gdcm::Anonymizer, 117
  - gdcm::DICOMDIRGenerator, 355
  - gdcm::FileDecompressLookupTable, 454
  - gdcm::FileDerivation, 458
  - gdcm::FileExplicitFilter, 461
  - gdcm::IconImageFilter, 511
  - gdcm::Printer, 855
  - gdcm::PythonFilter, 870
  - gdcm::Reader, 896
  - gdcm::SplitMosaicFilter, 1008
  - gdcm::StreamImageWriter, 1019
  - gdcm::StringFilter, 1039
  - gdcm::Validate, 1242
  - gdcm::Writer, 1397
  - gdcm::XMLPrinter, 1405
- SetFileName
  - gdcm::FileNameEvent, 478
  - gdcm::Reader, 896
  - gdcm::StreamImageReader, 1014
  - gdcm::StreamImageWriter, 1019
  - gdcm::Writer, 1398
  - vtkGDCMThreadedImageReader2, 1336
- SetFileNames
  - vtkGDCMImageReader, 1278
  - vtkGDCMImageWriter, 1306
  - vtkGDCMThreadedImageReader2, 1336
- SetFilePattern
  - vtkGDCMImageReader, 1278
  - vtkGDCMImageReader2, 1293
- SetFilePrefix
  - vtkGDCMImageReader, 1278
  - vtkGDCMImageReader2, 1294
- SetFilename
  - gdcm::TableReader, 1088
- SetFilenames
  - gdcm::DICOMDIRGenerator, 355
- SetFiles
  - gdcm::FileSet, 484
- SetFiniteVolume
  - gdcm::Surface, 1056
- SetForce
  - gdcm::ImageChangeTransferSyntax, 537
  - gdcm::ImageFragmentSplitter, 556
- SetForcePixelSpacing
  - gdcm::ImageHelper, 562
- SetForceRescaleInterceptSlope
  - gdcm::ImageHelper, 562
- SetFragmentSizeMax
  - gdcm::ImageFragmentSplitter, 556
- SetFrameOrigin
  - gdcm::Overlay, 764
- SetFromDataElement
  - gdcm::Attribute, 140
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 148
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 157
  - gdcm::Element, 395
  - gdcm::Element< TVR, VM::VM1\_n >, 402
- SetFromDataSet
  - gdcm::Attribute, 141
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 148
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, 157
  - gdcm::MediaStorage, 674
- SetFromFile
  - gdcm::MediaStorage, 674
- SetFromHeader
  - gdcm::MediaStorage, 674
- SetFromModality
  - gdcm::MediaStorage, 674
- SetFromSourceImageSequence
  - gdcm::MediaStorage, 675
- SetFromString
  - gdcm::DirectionCosines, 381
- SetFromUID
  - gdcm::UIDs, 1160
- SetGreenLUT
  - gdcm::LookupTable, 654
- SetGroup
  - gdcm::Curve, 305
  - gdcm::Overlay, 764
  - gdcm::Tag, 1099
- SetGroupXX
  - gdcm::DictEntry, 368
- SetHeader
  - gdcm::File, 443
- SetHighBit
  - gdcm::PixelFormat, 805
- SetHostname

- gdcm::ServiceClassUser, 976
- SetIconImage
  - gdcm::Pixmap, 811
- SetIE
  - gdcm::IODEntry, 592
- SetImage
  - gdcm::PixmapWriter, 823
  - gdcm::SplitMosaicFilter, 1008
- SetImplementationClassUID
  - gdcm::FileMetaInformation, 469
- SetImplementationVersionName
  - gdcm::FileMetaInformation, 470
- SetImplicitFlag
  - gdcm::network::ULConnectionCallback, 1210
- SetInput
  - gdcm::BitmapToBitmapFilter, 209
  - gdcm::ImageConverter, 554
  - vtkImageColorViewer, 1349
- SetInputConnection
  - vtkImageColorViewer, 1349
- SetInputDirectory
  - gdcm::EmptyMaskGenerator, 418
- SetInputFileName
  - gdcm::DictConverter, 364
  - gdcm::FileAnonymizer, 447
  - gdcm::FileChangeTransferSyntax, 451
- SetIntercept
  - gdcm::Image, 521
  - gdcm::Rescaler, 905
- SetKey
  - gdcm::CSAElement, 280
- SetKeyword
  - gdcm::DictEntry, 368
- SetLUT
  - gdcm::Bitmap, 202
  - gdcm::ImageCodec, 548
  - gdcm::LookupTable, 654
  - gdcm::SegmentedPaletteColorLookupTable, 937
- SetLastElement
  - gdcm::ParseException, 767
- SetLastFragment
  - gdcm::network::PresentationDataValue, 850
- SetLength
  - gdcm::ByteValue, 228
  - gdcm::Element< TVR, VM::VM1\_2 >, 397
  - gdcm::Element< TVR, VM::VM1\_n >, 402
  - gdcm::Element< TVR, VM::VM2\_2n >, 404
  - gdcm::Element< TVR, VM::VM2\_n >, 406
  - gdcm::Element< TVR, VM::VM3\_3n >, 408
  - gdcm::Element< TVR, VM::VM3\_n >, 410
  - gdcm::RLECodec, 911
  - gdcm::SequenceOfFragments, 952
  - gdcm::SequenceOfItems, 961
  - gdcm::Value, 1246
- SetLengthOnly
  - gdcm::ByteValue, 228
  - gdcm::Value, 1246
- SetLengthToUndefined
  - gdcm::SequenceOfItems, 961
- SetLoadMode
  - gdcm::SerieHelper, 967
- SetLookupTable
  - vtkImageMapToColors16, 1359
- SetLossless
  - gdcm::JPEGCodec, 629
  - gdcm::JPEGLSCodec, 637
- SetLossyError
  - gdcm::JPEGLSCodec, 637
- SetLossyFlag
  - gdcm::Bitmap, 201
  - gdcm::ImageCodec, 548
  - gdcm::PVRGCodec, 869
- SetManifold
  - gdcm::Surface, 1056
- SetMaxPDULength
  - gdcm::network::ULConnectionInfo, 1213
- SetMaxPDUSize
  - gdcm::network::ULConnection, 1207
- SetMaximumLength
  - gdcm::network::MaximumLengthSub, 663
- SetMaximumPointDistance
  - gdcm::Surface, 1056
- SetMeanPointDistance
  - gdcm::Surface, 1056
- SetMedicalImageProperties
  - vtkGDCMImageReader, 1279
  - vtkGDCMImageReader2, 1294
  - vtkGDCMImageWriter, 1306
  - vtkGDCMPolyDataWriter, 1323
- SetMergeModeToAbstractSyntax
  - gdcm::PresentationContextGenerator, 842
- SetMergeModeToTransferSyntax
  - gdcm::PresentationContextGenerator, 843
- SetMeshPrimitive
  - gdcm::Surface, 1056
- SetMessageHeader
  - gdcm::network::PresentationDataValue, 850
- SetMinMaxForPixelType
  - gdcm::Rescaler, 905
- SetName
  - gdcm::CSAElement, 281
  - gdcm::CSAHeaderDictEntry, 295
  - gdcm::DictEntry, 368
  - gdcm::IODEntry, 592
  - gdcm::Macro, 659
  - gdcm::Module, 695
  - gdcm::ModuleEntry, 699
  - gdcm::PDBElement, 778

- gdcm::network::AbstractSyntax, [106](#)
- gdcm::network::ApplicationContext, [120](#)
- gdcm::network::TransferSyntaxSub, [1125](#)
- SetNameFromUID
  - gdcm::network::AbstractSyntax, [106](#)
  - gdcm::network::TransferSyntaxSub, [1125](#)
- SetNeedByteSwap
  - gdcm::Bitmap, [202](#)
  - gdcm::ImageCodec, [549](#)
- SetNeedOverlayCleanup
  - gdcm::ImageCodec, [549](#)
- SetNestedDataSet
  - gdcm::Item, [604](#)
- SetNoOfItems
  - gdcm::CSAElement, [281](#)
- SetNoSwap
  - gdcm::Element, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [402](#)
- SetNumberOfCurves
  - gdcm::Pixmap, [811](#)
- SetNumberOfDimensions
  - gdcm::Bitmap, [202](#)
  - gdcm::ImageCodec, [549](#)
- SetNumberOfFilenames
  - gdcm::FilenameGenerator, [482](#)
- SetNumberOfFrames
  - gdcm::Overlay, [764](#)
- SetNumberOfInputPorts
  - vtkGDCMPolyDataWriter, [1323](#)
- SetNumberOfItems
  - gdcm::SequenceOfItems, [961](#)
- SetNumberOfOverlays
  - gdcm::Pixmap, [811](#)
- SetNumberOfPoints
  - gdcm::Curve, [305](#)
- SetNumberOfResolutions
  - gdcm::JPEG2000Codec, [618](#)
- SetNumberOfSegments
  - gdcm::SegmentWriter, [944](#)
- SetNumberOfSurfacePoints
  - gdcm::Surface, [1056](#)
- SetNumberOfSurfaces
  - gdcm::SurfaceWriter, [1069](#)
- SetNumberOfTableValues
  - vtkLookupTable16, [1377](#)
- SetNumberOfValues
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)
- SetNumberOfVectors
  - gdcm::Surface, [1057](#)
- SetObliquityThresholdCosineValue
  - gdcm::Orientation, [755](#)
- SetOffScreenRendering
  - vtkImageColorViewer, [1349](#)
- SetOrigin
  - gdcm::Image, [521](#), [522](#)
  - gdcm::Overlay, [764](#)
- SetOriginValue
  - gdcm::ImageHelper, [563](#)
- SetOutputDimensions
  - gdcm::IconImageGenerator, [514](#)
- SetOutputDirectory
  - gdcm::EmptyMaskGenerator, [419](#)
- SetOutputFileName
  - gdcm::DictConverter, [364](#)
  - gdcm::FileAnonymizer, [447](#)
  - gdcm::FileChangeTransferSyntax, [451](#)
  - gdcm::FileStreamer, [489](#)
- SetOutputFormatToLuminance
  - vtkImageMapToColors16, [1360](#)
- SetOutputFormatToLuminanceAlpha
  - vtkImageMapToColors16, [1360](#)
- SetOutputFormatToRGBA
  - vtkImageMapToColors16, [1360](#)
- SetOutputFormatToRGB
  - vtkImageMapToColors16, [1360](#)
- SetOutputType
  - gdcm::DictConverter, [364](#)
- SetOutsideValuePixel
  - gdcm::IconImageGenerator, [514](#)
- SetOverlay
  - gdcm::Overlay, [765](#)
- SetOverlayVisibility
  - vtkImageColorViewer, [1349](#)
- SetOwner
  - gdcm::PrivateTag, [862](#)
- SetPDU
  - gdcm::network::ULEvent, [1223](#)
- SetPMSRescaleInterceptSlope
  - gdcm::ImageHelper, [563](#)
- SetParentId
  - vtkImageColorViewer, [1349](#)
- SetPassword
  - gdcm::CAPICryptographicMessageSyntax, [234](#)
  - gdcm::CryptographicMessageSyntax, [275](#)
  - gdcm::OpenSSLCryptographicMessageSyntax, [746](#)
  - gdcm::OpenSSL7CryptographicMessageSyntax, [751](#)
- SetPattern
  - gdcm::FilenameGenerator, [482](#)
- SetPermissions
  - gdcm::System, [1080](#)
- SetPhotometricInterpretation
  - gdcm::Bitmap, [202](#)
  - gdcm::ImageChangePhotometricInterpretation, [529](#)
  - gdcm::ImageCodec, [549](#)
- SetPixelFormat
  - gdcm::Bitmap, [202](#)

- gdcm::ImageCodec, [549](#)
- gdcm::JPEGCodec, [630](#)
- gdcm::Rescaler, [905](#)
- SetPixelMinMax
  - gdcm::IconImageGenerator, [514](#)
- SetPixelRepresentation
  - gdcm::PixelFormat, [805](#)
- SetPixmap
  - gdcm::FileDecompressLookupTable, [454](#)
  - gdcm::IconImageGenerator, [515](#)
  - gdcm::PixmapWriter, [824](#)
- SetPlanarConfiguration
  - gdcm::Bitmap, [203](#)
  - gdcm::ImageChangePlanarConfiguration, [533](#)
  - gdcm::ImageCodec, [550](#)
- SetPointCoordinatesData
  - gdcm::Surface, [1057](#)
- SetPointPositionAccuracy
  - gdcm::Surface, [1057](#)
- SetPointsBoundingBoxCoordinates
  - gdcm::Surface, [1057](#)
- SetPort
  - gdcm::ServiceClassUser, [976](#)
- SetPortSCP
  - gdcm::ServiceClassUser, [976](#)
- SetPosition
  - vtkImageColorViewer, [1349](#), [1350](#)
- SetPreamble
  - gdcm::FileMetaInformation, [470](#)
- SetPrefix
  - gdcm::FilenameGenerator, [482](#)
- SetPresentationContextID
  - gdcm::PresentationContext, [836](#)
  - gdcm::network::PresentationContextAC, [838](#)
  - gdcm::network::PresentationContextRQ, [846](#)
  - gdcm::network::PresentationDataValue, [851](#)
- SetPresentationContexts
  - gdcm::ServiceClassUser, [977](#)
  - gdcm::network::ULConnection, [1207](#)
- SetPrettyPrint
  - gdcm::JSON, [640](#)
- SetPrimitiveData
  - gdcm::MeshPrimitive, [685](#)
- SetPrimitiveType
  - gdcm::MeshPrimitive, [685](#)
- SetPrimitivesData
  - gdcm::MeshPrimitive, [685](#)
- SetPrivateCreator
  - gdcm::Tag, [1100](#)
- SetProcessingAlgorithm
  - gdcm::Surface, [1057](#)
- SetProgress
  - gdcm::ProgressEvent, [865](#)
- SetPropertyCategory
  - gdcm::Segment, [932](#)
- SetPropertyType
  - gdcm::Segment, [932](#)
- SetPropertyTypeModifiers
  - gdcm::Segment, [932](#)
- SetPurposeOfReferenceCodeSequenceCodeValue
  - gdcm::FileDerivation, [458](#)
- SetQuality
  - gdcm::JPEG2000Codec, [618](#)
  - gdcm::JPEGCodec, [630](#)
- SetRTStructSetProperties
  - vtkGDCMPolyDataWriter, [1323](#)
- SetRate
  - gdcm::JPEG2000Codec, [618](#)
- SetReason
  - gdcm::network::AAAbortPDU, [89](#)
  - gdcm::network::PresentationContextAC, [839](#)
- SetRecommendedDisplayCIELabValue
  - gdcm::Surface, [1057](#), [1058](#)
- SetRecommendedDisplayGrayscaleValue
  - gdcm::Surface, [1058](#)
- SetRecommendedPresentationOpacity
  - gdcm::Surface, [1058](#)
- SetRecommendedPresentationType
  - gdcm::Surface, [1058](#)
- SetRecomputeItemLength
  - gdcm::FileExplicitFilter, [461](#)
- SetRecomputeSequenceLength
  - gdcm::FileExplicitFilter, [461](#)
- SetRedLUT
  - gdcm::LookupTable, [654](#)
- SetRef
  - gdcm::IODEntry, [592](#)
- SetRegion
  - gdcm::ImageRegionReader, [572](#)
- SetRenderWindow
  - vtkImageColorViewer, [1350](#)
- SetRenderer
  - vtkImageColorViewer, [1350](#)
- SetRescaleInterceptSlopeValue
  - gdcm::ImageHelper, [563](#)
- SetRetired
  - gdcm::DictEntry, [369](#)
- SetReversible
  - gdcm::JPEG2000Codec, [619](#)
- SetRoot
  - gdcm::UIDGenerator, [1134](#)
- SetRootDirectory
  - gdcm::DICOMDIRGenerator, [356](#)
- SetRows
  - gdcm::Bitmap, [203](#)
  - gdcm::Overlay, [765](#)
- SetSOPClassUIDMode
  - gdcm::EmptyMaskGenerator, [419](#)

- SetSOPInstanceUID
  - gdcm::BaseQuery, [180](#)
- SetSamplesPerPixel
  - gdcm::PixelFormat, [806](#)
- SetScalarType
  - gdcm::PixelFormat, [806](#)
- SetSearchParameter
  - gdcm::BaseQuery, [179](#)
- SetSegmentAlgorithmName
  - gdcm::Segment, [932](#)
- SetSegmentAlgorithmType
  - gdcm::Segment, [932](#)
- SetSegmentDescription
  - gdcm::Segment, [933](#)
- SetSegmentLabel
  - gdcm::Segment, [933](#)
- SetSegmentNumber
  - gdcm::Segment, [933](#)
- SetSegments
  - gdcm::SegmentWriter, [945](#)
- SetSize
  - vtkImageColorViewer, [1350](#)
- SetSlice
  - vtkImageColorViewer, [1351](#)
- SetSliceOrientation
  - vtkImageColorViewer, [1351](#)
- SetSliceOrientationToXY
  - vtkImageColorViewer, [1351](#)
- SetSliceOrientationToXZ
  - vtkImageColorViewer, [1351](#)
- SetSliceOrientationToYZ
  - vtkImageColorViewer, [1351](#)
- SetSlope
  - gdcm::Image, [522](#)
  - gdcm::Rescaler, [905](#)
- SetSortFunction
  - gdcm::Sorter, [1001](#)
- SetSource
  - gdcm::network::AAabortPDU, [89](#)
- SetSourceApplicationEntityTitle
  - gdcm::FileMetaInformation, [470](#)
- SetSpacing
  - gdcm::Image, [522](#)
- SetSpacingValue
  - gdcm::ImageHelper, [563](#)
- SetState
  - gdcm::network::ULConnection, [1207](#)
- SetStream
  - gdcm::Reader, [896](#)
  - gdcm::StreamImageReader, [1014](#)
  - gdcm::StreamImageWriter, [1019](#)
  - gdcm::Trace, [1116](#)
  - gdcm::Writer, [1398](#)
- SetStreamToFile
  - gdcm::Trace, [1116](#)
- SetStyle
  - gdcm::Printer, [855](#)
  - gdcm::XMLPrinter, [1406](#)
- SetSurfaceComments
  - gdcm::Surface, [1058](#)
- SetSurfaceCount
  - gdcm::Segment, [933](#)
- SetSurfaceNumber
  - gdcm::Surface, [1059](#)
- SetSurfaceProcessing
  - gdcm::Surface, [1059](#)
- SetSurfaceProcessingDescription
  - gdcm::Surface, [1059](#)
- SetSurfaceProcessingRatio
  - gdcm::Surface, [1059](#)
- SetSyngoDT
  - gdcm::CSAElement, [281](#)
- SetTag
  - gdcm::AnonymizeEvent, [109](#)
  - gdcm::DataElement, [317](#)
- SetTagsToRead
  - gdcm::Sorter, [1001](#)
- SetTargetPixelType
  - gdcm::Rescaler, [905](#)
- SetTemplateFileName
  - gdcm::FileStreamer, [489](#)
- SetTileSize
  - gdcm::JPEG2000Codec, [619](#)
- SetTimeout
  - gdcm::ServiceClassUser, [977](#)
  - gdcm::network::ARTIMTimer, [130](#)
- SetToUndefined
  - gdcm::VL, [1254](#)
- SetTransferSyntax
  - gdcm::Bitmap, [203](#)
  - gdcm::FileChangeTransferSyntax, [451](#)
  - gdcm::ImageChangeTransferSyntax, [537](#)
  - gdcm::network::PresentationContextAC, [839](#)
- SetTuple
  - gdcm::network::RoleSelectionSub, [913](#)
  - gdcm::network::SOPClassExtendedNegociationSub, [995](#)
  - gdcm::network::ServiceClassApplicationInformation, [969](#)
- SetType
  - gdcm::ModuleEntry, [699](#)
  - gdcm::Overlay, [765](#)
- SetTypeOfData
  - gdcm::Curve, [305](#)
- SetUsage
  - gdcm::IODEntry, [592](#)
- SetUseSeriesDetails
  - gdcm::SerieHelper, [967](#)



- SetUseTargetPixelType
  - gdcm::Rescaler, [906](#)
- SetUseVRUN
  - gdcm::FileExplicitFilter, [461](#)
- SetUserCodec
  - gdcm::ImageChangeTransferSyntax, [538](#)
- SetUserData
  - gdcm::Parser, [772](#)
- SetUserInformation
  - gdcm::network::AAAssociateRQPDU, [102](#)
- SetVLToUndefined
  - gdcm::DataElement, [318](#)
- SetValue
  - gdcm::Attribute, [141](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [148](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [157](#)
  - gdcm::CSAElement, [281](#)
  - gdcm::DataElement, [317](#)
  - gdcm::Element, [395](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [402](#)
  - gdcm::PDBelement, [778](#)
- SetValueFieldLength
  - gdcm::DataElement, [318](#)
- SetValues
  - gdcm::Attribute, [141](#)
  - gdcm::Attribute< Group, Element, TVR, VM::VM1\_n >, [158](#)
- SetVectorAccuracy
  - gdcm::Surface, [1059](#)
- SetVectorCoordinateData
  - gdcm::Surface, [1059](#)
- SetVectorDimensionality
  - gdcm::Surface, [1060](#)
- SetVL
  - gdcm::DataElement, [318](#)
- SetVM
  - gdcm::CSAElement, [281](#)
  - gdcm::CSAHeaderDictEntry, [295](#)
  - gdcm::DictEntry, [369](#)
- SetVR
  - gdcm::CSAElement, [281](#)
  - gdcm::CSAHeaderDictEntry, [295](#)
  - gdcm::DataElement, [318](#)
  - gdcm::DictEntry, [369](#)
- SetWarning
  - gdcm::Trace, [1116](#)
- SetWarningStream
  - gdcm::Trace, [1116](#)
- SetWindowId
  - vtkImageColorViewer, [1352](#)
- SetWriteDataSetOnly
  - gdcm::Writer, [1398](#)
- SetZSpacingTolerance
  - gdcm::IPPSorter, [599](#)
- setAttribute
  - gdcm::terminal, [86](#)
- setbgcolor
  - gdcm::terminal, [86](#)
- setfgcolor
  - gdcm::terminal, [86](#)
- setmode
  - gdcm::terminal, [86](#)
- SetupInteractor
  - vtkImageColorViewer, [1351](#)
- Shift
  - vtkGDCMImageReader, [1287](#)
  - vtkGDCMImageReader2, [1302](#)
- ShiftEnd
  - gdcm::ByteBuffer, [216](#)
- ShowAbort
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowAnonymization
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowData
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowDataSet
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowFileName
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowIteration
  - gdcm::SimpleSubjectWatcher, [986](#)
- ShowProgress
  - gdcm::SimpleSubjectWatcher, [987](#)
- SimpleMemberCommand
  - gdcm::SimpleMemberCommand, [982](#)
- SimpleSubjectWatcher
  - gdcm::SimpleSubjectWatcher, [985](#)
- SingleSerieUIDFileSetHT
  - gdcm::SerieHelper, [968](#)
- SingleSerieUIDFileSetmap
  - gdcm::SerieHelper, [964](#)
- Size
  - gdcm::CodeString, [253](#)
  - gdcm::DataSet, [336](#)
  - gdcm::GroupDict, [507](#)
  - gdcm::network::AAAbortPDU, [89](#)
  - gdcm::network::AAAssociateACPDU, [93](#)
  - gdcm::network::AAAssociateRJPDPU, [96](#)
  - gdcm::network::AAAssociateRQPDU, [102](#)
  - gdcm::network::AReleaseRPPDU, [126](#)
  - gdcm::network::AReleaseRQPDU, [128](#)
  - gdcm::network::AbstractSyntax, [106](#)
  - gdcm::network::ApplicationContext, [120](#)
  - gdcm::network::AsynchronousOperationsWindow↔  
Sub, [133](#)
  - gdcm::network::BasePDU, [175](#)

- gdcmm::network::ImplementationClassUIDSub, [579](#)
- gdcmm::network::ImplementationVersionNameSub, [582](#)
- gdcmm::network::MaximumLengthSub, [663](#)
- gdcmm::network::PDataTFPDU, [775](#)
- gdcmm::network::PresentationContextAC, [839](#)
- gdcmm::network::PresentationContextRQ, [847](#)
- gdcmm::network::PresentationDataValue, [851](#)
- gdcmm::network::RoleSelectionSub, [913](#)
- gdcmm::network::SOPClassExtendedNegociationSub, [995](#)
- gdcmm::network::ServiceClassApplicationInformation, [970](#)
- gdcmm::network::TransferSyntaxSub, [1125](#)
- gdcmm::network::UserInformation, [1239](#)
- size\_type
  - gdcmm::CodeString, [252](#)
  - gdcmm::LO, [646](#)
  - gdcmm::String, [1035](#)
- SizeType
  - gdcmm::DataSet, [328](#)
  - gdcmm::FilenameGenerator, [480](#)
  - gdcmm::IOD, [588](#)
  - gdcmm::NestedModuleEntries, [722](#)
  - gdcmm::PresentationContext, [834](#)
  - gdcmm::PresentationContextGenerator, [841](#)
  - gdcmm::SequenceOfFragments, [948](#)
  - gdcmm::SequenceOfItems, [956](#)
  - gdcmm::network::AAssociateACPDU, [91](#)
  - gdcmm::network::AAssociateRQPDU, [99](#)
  - gdcmm::network::PDataTFPDU, [774](#)
  - gdcmm::network::PresentationContextRQ, [844](#)
- Slice
  - vtkImageColorViewer, [1356](#)
- SliceOrientation
  - vtkImageColorViewer, [1356](#)
- Slices
  - gdcmm::MrProtocol::SliceArray, [989](#)
- SmartPointer
  - gdcmm::Object, [742](#)
  - gdcmm::SmartPointer, [991](#), [992](#)
- Sort
  - gdcmm::IPPSorter, [599](#)
  - gdcmm::Sorter, [1001](#)
- SortFunc
  - gdcmm::Sorter, [1003](#)
- SortFunction
  - gdcmm::Sorter, [1000](#)
- Sorter
  - gdcmm::Sorter, [1000](#)
- Spacing
  - gdcmm::Spacing, [1005](#)
- SpacingType
  - gdcmm::Spacing, [1004](#)
- Spectroscopy
  - gdcmm::Spectroscopy, [1006](#)
- Split
  - gdcmm::ImageFragmentSplitter, [557](#)
  - gdcmm::SplitMosaicFilter, [1009](#)
- SplitExtent
  - vtkGDCMThreadedImageReader2, [1336](#)
- SplitMosaicFilter
  - gdcmm::SplitMosaicFilter, [1007](#)
- Squeeze
  - gdcmm::ApplicationEntity, [123](#)
- StableSort
  - gdcmm::Sorter, [1002](#)
- Start
  - gdcmm::network::ARTIMTimer, [130](#)
- StartAssociation
  - gdcmm::ServiceClassUser, [977](#)
- StartDataElement
  - gdcmm::FileStreamer, [489](#)
- StartElement
  - gdcmm::TableReader, [1088](#)
  - gdcmm::XMLDictReader, [1402](#)
  - gdcmm::XMLPrivateDictReader, [1409](#)
- StartElementHandler
  - gdcmm::Parser, [769](#)
- StartEncode
  - gdcmm::ImageCodec, [550](#)
  - gdcmm::JPEG2000Codec, [619](#)
  - gdcmm::JPEGCodec, [630](#)
  - gdcmm::JPEGLSCodec, [637](#)
  - gdcmm::RLECodec, [911](#)
- StartFilter
  - gdcmm::SimpleSubjectWatcher, [987](#)
- StartGroupDataElement
  - gdcmm::FileStreamer, [489](#)
- Stop
  - gdcmm::network::ARTIMTimer, [130](#)
- StopAssociation
  - gdcmm::ServiceClassUser, [977](#)
- StopDataElement
  - gdcmm::FileStreamer, [490](#)
- StopEncode
  - gdcmm::ImageCodec, [550](#)
  - gdcmm::JPEG2000Codec, [619](#)
  - gdcmm::JPEGCodec, [630](#)
  - gdcmm::JPEGLSCodec, [638](#)
  - gdcmm::RLECodec, [911](#)
- StopGroupDataElement
  - gdcmm::FileStreamer, [490](#)
- StopProtocol
  - gdcmm::network::ULConnection, [1207](#)
- StrCaseCmp
  - gdcmm::System, [1080](#)
- StrNCaseCmp



- gdcm::System, [1080](#)
- StrSep
  - gdcm::System, [1081](#)
- StrTokR
  - gdcm::System, [1081](#)
- Stream
  - gdcm::Writer, [1399](#)
- StreamImageReader
  - gdcm::Reader, [897](#)
  - gdcm::StreamImageReader, [1012](#)
- StreamImageWriter
  - gdcm::StreamImageWriter, [1017](#)
  - gdcm::Writer, [1399](#)
- StrictScanner
  - gdcm::StrictScanner, [1026](#)
- String
  - gdcm::String, [1035](#), [1036](#)
- StringFilter
  - gdcm::StringFilter, [1038](#)
- StructureSetDate
  - vtkRTStructSetProperties, [1388](#)
- StructureSetLabel
  - vtkRTStructSetProperties, [1389](#)
- StructureSetName
  - vtkRTStructSetProperties, [1389](#)
- StructureSetTime
  - vtkRTStructSetProperties, [1389](#)
- Study
  - gdcm::Study, [1041](#)
- StudyInstanceUID
  - vtkRTStructSetProperties, [1389](#)
- Subject
  - gdcm::Subject, [1043](#)
- Superclass
  - gdcm::AnonymizeEvent, [108](#)
  - gdcm::DataEvent, [323](#)
  - gdcm::DataSetEvent, [339](#)
  - gdcm::FileNameEvent, [477](#)
  - gdcm::LO, [646](#)
  - gdcm::ProgressEvent, [864](#)
- Surface
  - gdcm::Surface, [1049](#)
- SurfaceCount
  - gdcm::Segment, [935](#)
- SurfaceReader
  - gdcm::SurfaceReader, [1065](#)
- SurfaceVector
  - gdcm::Segment, [927](#)
- SurfaceWriter
  - gdcm::SurfaceWriter, [1068](#)
- Surfaces
  - gdcm::Segment, [935](#)
- Swap
  - gdcm::ByteSwap, [217](#)
  - gdcm::SwapperDoOp, [1073](#)
  - gdcm::SwapperNoOp, [1073](#)
- SwapArray
  - gdcm::SwapperDoOp, [1073](#)
  - gdcm::SwapperNoOp, [1074](#)
- SwapCode
  - gdcm::SwapCode, [1071](#)
- SwapCodeType
  - gdcm::SwapCode, [1071](#)
- SwapFromSwapCodeIntoSystem
  - gdcm::ByteSwap, [218](#)
- SwapRange
  - gdcm::ByteSwap, [218](#)
- SwapRangeFromSwapCodeIntoSystem
  - gdcm::ByteSwap, [218](#)
- SyngoDTField
  - gdcm::CSAElement, [283](#)
- SystemIsBigEndian
  - gdcm::ByteSwap, [218](#)
- SystemIsLittleEndian
  - gdcm::ByteSwap, [219](#)
- TConstMemberFunctionPointer
  - gdcm::MemberCommand, [677](#)
- TMComp
  - gdcm, [60](#)
- TMemberFunctionPointer
  - gdcm::MemberCommand, [678](#)
  - gdcm::SimpleMemberCommand, [982](#)
- TSName
  - gdcm::UIDs, [1145](#)
- TSType
  - gdcm::TransferSyntax, [1120](#)
  - gdcm::UIDs, [1152](#)
- TYPETOENCODING
  - gdcm, [76](#)
  - gdcmVR.h, [1678](#)
- TYPETOLENGTH
  - gdcmVM.h, [1676](#)
- Table
  - gdcm::Table, [1082](#)
- Table16
  - vtkLookupTable16, [1377](#)
- TableEntry
  - gdcm::TableEntry, [1084](#)
- TableReader
  - gdcm::TableReader, [1085](#)
- TableRow
  - gdcm::network::TableRow, [1089](#)
- Tag
  - gdcm::Tag, [1092](#), [1093](#)
- tag
  - gdcm::Tag, [1101](#)
- TagField

- gdcm::DataElement, [319](#)
- TagPath
  - gdcm::TagPath, [1102](#)
- TagToValue
  - gdcm::Scanner, [918](#)
  - gdcm::StrictScanner, [1026](#)
- TagToValueValueType
  - gdcm::Scanner, [918](#)
  - gdcm::StrictScanner, [1026](#)
- tags
  - gdcm::Tag, [1101](#)
- TagsToRead
  - gdcm::Sorter, [1003](#)
- TestAbortOff
  - gdcm::SimpleSubjectWatcher, [987](#)
- TestAbortOn
  - gdcm::SimpleSubjectWatcher, [987](#)
- TestPBKDF2
  - gdcm::ASN1, [132](#)
- Testing
  - gdcm::Testing, [1106](#)
- TestsList.txt, [1684](#)
- ThreadedExecute
  - vtkImageRGBToYBR, [1371](#)
  - vtkImageYBRToRGB, [1373](#)
- ThreadedRequestData
  - vtkGDCMThreadedImageReader2, [1336](#)
  - vtkImageMapToColors16, [1360](#)
  - vtkImageMapToWindowLevelColors2, [1365](#)
- to\_string
  - gdcm, [76](#)
- ToPyObject
  - gdcm::PythonFilter, [870](#)
- ToString
  - gdcm::StringFilter, [1040](#)
- ToStringPair
  - gdcm::StringFilter, [1040](#), [1041](#)
- ToUnixSlashes
  - gdcm::Filename, [474](#)
- ToWindowsSlashes
  - gdcm::Filename, [474](#)
- Trace
  - gdcm::Trace, [1113](#)
- TransferSyntax
  - gdcm::TransferSyntax, [1120](#)
- TransferSyntaxArrayType
  - gdcm::PresentationContext, [834](#)
- TransferSyntaxStringsType
  - gdcm::UIDs, [1145](#)
- TransferSyntaxSub
  - gdcm::network::TransferSyntaxSub, [1124](#)
- TransferSyntaxes
  - gdcm::PresentationContext, [836](#)
- Transition
  - gdcm::network::Transition, [1127](#)
- transitions
  - gdcm::network::TableRow, [1090](#)
- Trim
  - gdcm::String, [1036](#), [1037](#)
- TrimInternal
  - gdcm::CodeString, [253](#)
- Truncate
  - gdcm::String, [1037](#)
- TryJPEG2000Codec
  - gdcm::Bitmap, [203](#)
  - gdcm::ImageChangeTransferSyntax, [538](#)
- TryJPEG2000Codec2
  - gdcm::Bitmap, [204](#)
- TryJPEGCodec
  - gdcm::Bitmap, [204](#)
  - gdcm::ImageChangeTransferSyntax, [538](#)
- TryJPEGCodec2
  - gdcm::Bitmap, [204](#)
- TryJPEGLSCodec
  - gdcm::Bitmap, [204](#)
  - gdcm::ImageChangeTransferSyntax, [538](#)
- TryKAKADUCodec
  - gdcm::Bitmap, [204](#)
- TryPVRGCodec
  - gdcm::Bitmap, [204](#)
- TryRAWCodec
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageChangeTransferSyntax, [539](#)
- TryRLECodec
  - gdcm::Bitmap, [205](#)
  - gdcm::ImageChangeTransferSyntax, [539](#)
- TS
  - gdcm::Bitmap, [207](#)
- Type
  - gdcm::Element, [393](#)
  - gdcm::Element< TVR, VM::VM1\_n >, [399](#)
  - gdcm::Type, [1130](#)
  - gdcm::VL, [1252](#)
- TypeType
  - gdcm::Type, [1129](#)
- UIComp
  - gdcm, [60](#)
- UIDGenerator
  - gdcm::UIDGenerator, [1133](#)
- ULAction
  - gdcm::network::ULAction, [1162](#)
- ULActionAE6
  - gdcm::network::ULConnection, [1208](#)
- ULBasicCallback
  - gdcm::network::ULBasicCallback, [1202](#)
- ULConnection
  - gdcm::network::ULConnection, [1204](#)

- ULConnectionCallback
  - gdcm::network::ULConnectionCallback, [1209](#)
- ULConnectionInfo
  - gdcm::network::ULConnectionInfo, [1211](#)
- ULConnectionManager
  - gdcm::network::ULConnection, [1208](#)
  - gdcm::network::ULConnectionManager, [1215](#)
- ULError
  - gdcm::network::ULError, [1221](#), [1222](#)
- ULTransitionTable
  - gdcm::network::ULTransitionTable, [1224](#)
- ULWritingCallback
  - gdcm::network::ULWritingCallback, [1226](#)
- UTComp
  - gdcm, [60](#)
- UnInstallPipeline
  - vtkImageColorViewer, [1352](#)
- UnRegister
  - gdcm::Object, [741](#)
- Unpack
  - gdcm::Unpacker12Bits, [1233](#)
- UnusedBitsPresentInPixelData
  - gdcm::Bitmap, [205](#)
  - gdcm::Pixmap, [812](#)
- Update
  - gdcm::Curve, [305](#)
  - gdcm::Overlay, [765](#)
- UpdateDisplayExtent
  - vtkImageColorViewer, [1352](#)
- UpdateOrientation
  - vtkImageColorViewer, [1352](#)
- UpdatePosition
  - gdcm::ByteBuffer, [216](#)
- Usage
  - gdcm::Usage, [1235](#)
- UsageType
  - gdcm::Usage, [1234](#)
- UseDictAlways
  - gdcm::PythonFilter, [871](#)
  - gdcm::StringFilter, [1041](#)
- UserInfo
  - gdcm::network::UserInfo, [1238](#)
- UserOrdering
  - gdcm::SerieHelper, [967](#)
- V
  - gdcm::Validate, [1243](#)
- VIEWType
  - gdcm::Surface, [1048](#)
- VMType
  - gdcm::VM, [1257](#)
- VRBINARY
  - gdcm, [77](#)
- VRDS16ILLEGAL
  - gdcmElement.h, [1479](#)
- VRField
  - gdcm::CSAElement, [283](#)
  - gdcm::DataElement, [320](#)
- VRType
  - gdcm::VR, [1262](#)
- VRTypeTemplateCase
  - gdcmVR.h, [1678](#)
- VTK\_CMYK
  - vtkGDCMImageReader.h, [1685](#)
  - vtkGDCMImageReader2.h, [1686](#)
- VTK\_INVERSE\_LUMINANCE
  - vtkGDCMImageReader.h, [1685](#)
  - vtkGDCMImageReader2.h, [1686](#)
- VTK\_LEGACY
  - vtkImageColorViewer, [1352](#), [1353](#)
- VTK\_LOOKUP\_TABLE
  - vtkGDCMImageReader.h, [1685](#)
  - vtkGDCMImageReader2.h, [1687](#)
- VTK\_YBR
  - vtkGDCMImageReader.h, [1685](#)
  - vtkGDCMImageReader2.h, [1687](#)
- Valid
  - gdcm::Preamble, [832](#)
- ValidDataSet
  - gdcm::BaseQuery, [180](#)
- Validate
  - gdcm::PixelFormat, [806](#)
  - gdcm::Validate, [1242](#)
- ValidateQuery
  - gdcm::BaseQuery, [180](#)
  - gdcm::BaseRootQuery, [185](#)
  - gdcm::FindPatientRootQuery, [494](#)
  - gdcm::FindStudyRootQuery, [497](#)
  - gdcm::ModalityPerformedProcedureStepCreateQuery, [688](#)
  - gdcm::ModalityPerformedProcedureStepSetQuery, [691](#)
  - gdcm::MovePatientRootQuery, [705](#)
  - gdcm::MoveStudyRootQuery, [708](#)
  - gdcm::WLMFindQuery, [1392](#)
- Validation
  - gdcm::Validate, [1243](#)
- Value
  - gdcm::Value, [1245](#)
- value
  - gdcm::SerieHelper::Rule, [915](#)
- value\_type
  - gdcm::CodeString, [252](#)
  - gdcm::LO, [646](#)
  - gdcm::String, [1035](#)
- ValueField
  - gdcm::DataElement, [320](#)
  - gdcm::PDBelement, [779](#)

- ValueLengthField
  - gdcm::DataElement, 320
- ValueMultiplicityField
  - gdcm::CSAElement, 283
- ValuePtr
  - gdcm::DataElement, 309
- ValueType
  - gdcm::Scanner, 919
  - gdcm::StrictScanner, 1026
- Verify
  - gdcm::Defs, 348
  - gdcm::Macro, 659
  - gdcm::Module, 695
- Version
  - gdcm::Version, 1249
- VL
  - gdcm::VL, 1252
- VM
  - gdcm::VM, 1258
- VR
  - gdcm::VR, 1263
- vtkBooleanMacro
  - vtkGDCMImageReader, 1279
  - vtkGDCMImageReader2, 1294, 1295
  - vtkGDCMImageWriter, 1307
  - vtkGDCMThreadedImageReader, 1332
  - vtkGDCMThreadedImageReader2, 1337
  - vtkImageColorViewer, 1353
  - vtkImageMapToColors16, 1360
- vtkGDCMImageReader, 1273
  - ~vtkGDCMImageReader, 1275
  - ApplyInverseVideo, 1284
  - ApplyLookupTable, 1284
  - ApplyPlanarConfiguration, 1284
  - ApplyShiftScale, 1284
  - ApplyYBRToRGB, 1284
  - CanReadFile, 1276
  - Curve, 1285
  - DirectionCosines, 1285
  - ExecuteData, 1276
  - ExecuteInformation, 1276
  - FileNames, 1285
  - FillMedicalImageInformation, 1276
  - ForceRescale, 1285
  - GetDescriptiveName, 1276
  - GetFileExtensions, 1276
  - GetIconImage, 1277
  - GetOverlay, 1277
  - IconDataScalarType, 1285
  - IconImageDataExtent, 1285
  - IconNumberOfScalarComponents, 1285
  - ImageFormat, 1286
  - ImageOrientationPatient, 1286
  - ImagePositionPatient, 1286
  - LoadIconImage, 1286
  - LoadOverlays, 1286
  - LoadSingleFile, 1277
  - LossyFlag, 1286
  - MedicalImageProperties, 1286
  - New, 1277
  - NumberOfIconImages, 1287
  - NumberOfOverlays, 1287
  - PlanarConfiguration, 1287
  - PrintSelf, 1277
  - RequestDataCompat, 1278
  - RequestInformationCompat, 1278
  - Scale, 1287
  - SetCurve, 1278
  - SetFileNames, 1278
  - SetFilePattern, 1278
  - SetFilePrefix, 1278
  - SetMedicalImageProperties, 1279
  - Shift, 1287
  - vtkBooleanMacro, 1279
  - vtkGDCMImageReader, 1275
  - vtkGDCMMedicalImageProperties, 1314
  - vtkGetMacro, 1280, 1281
  - vtkGetObjectMacro, 1281, 1282
  - vtkGetStringMacro, 1282
  - vtkGetVector3Macro, 1282
  - vtkGetVector6Macro, 1283
  - vtkSetMacro, 1283
  - vtkSetVector6Macro, 1283
  - vtkTypeMacro, 1284
- vtkGDCMImageReader.h, 1684
  - VTK\_CMYK, 1685
  - VTK\_INVERSE\_LUMINANCE, 1685
  - VTK\_LOOKUP\_TABLE, 1685
  - VTK\_YBR, 1685
- vtkGDCMImageReader2, 1288
  - ~vtkGDCMImageReader2, 1290
  - ApplyInverseVideo, 1299
  - ApplyLookupTable, 1299
  - ApplyPlanarConfiguration, 1299
  - ApplyShiftScale, 1299
  - ApplyYBRToRGB, 1299
  - CanReadFile, 1291
  - Curve, 1299
  - DirectionCosines, 1300
  - FillMedicalImageInformation, 1291
  - ForceRescale, 1300
  - GetDescriptiveName, 1291
  - GetFileExtensions, 1291
  - GetIconImage, 1291
  - GetIconImagePort, 1291
  - GetOverlay, 1291
  - GetOverlayPort, 1292
  - IconDataScalarType, 1300

- IconImageDataExtent, [1300](#)
- IconNumberOfScalarComponents, [1300](#)
- ImageFormat, [1300](#)
- ImageOrientationPatient, [1300](#)
- ImagePositionPatient, [1301](#)
- LoadIconImage, [1301](#)
- LoadOverlays, [1301](#)
- LoadSingleFile, [1292](#)
- LossyFlag, [1301](#)
- New, [1292](#)
- NumberOfIconImages, [1301](#)
- NumberOfOverlays, [1301](#)
- PlanarConfiguration, [1301](#)
- PrintSelf, [1292](#)
- ProcessRequest, [1292](#)
- RequestData, [1293](#)
- RequestDataCompat, [1293](#)
- RequestInformation, [1293](#)
- RequestInformationCompat, [1293](#)
- Scale, [1302](#)
- SetCurve, [1293](#)
- SetFilePattern, [1293](#)
- SetFilePrefix, [1294](#)
- SetMedicalImageProperties, [1294](#)
- Shift, [1302](#)
- vtkBooleanMacro, [1294](#), [1295](#)
- vtkGDCMImageReader2, [1290](#)
- vtkGDCMMedicalImageProperties, [1314](#)
- vtkGetMacro, [1295](#), [1296](#)
- vtkGetObjectMacro, [1297](#)
- vtkGetStringMacro, [1297](#)
- vtkGetVector3Macro, [1297](#)
- vtkGetVector6Macro, [1297](#)
- vtkSetMacro, [1298](#)
- vtkSetVector6Macro, [1298](#)
- vtkTypeMacro, [1298](#)
- vtkGDCMImageReader2.h, [1686](#)
  - VTK\_CMYK, [1686](#)
  - VTK\_INVERSE\_LUMINANCE, [1686](#)
  - VTK\_LOOKUP\_TABLE, [1687](#)
  - VTK\_YBR, [1687](#)
- vtkGDCMImageWriter, [1302](#)
  - ~vtkGDCMImageWriter, [1305](#)
  - CompressionTypes, [1304](#)
  - GetDescriptiveName, [1305](#)
  - GetFileExtensions, [1305](#)
  - GetFileName, [1305](#)
  - New, [1305](#)
  - PrintSelf, [1306](#)
  - SetDirectionCosines, [1306](#)
  - SetDirectionCosinesFromImageOrientationPatient, [1306](#)
  - SetFileNames, [1306](#)
  - SetMedicalImageProperties, [1306](#)
  - vtkBooleanMacro, [1307](#)
  - vtkGDCMImageWriter, [1305](#)
  - vtkGDCMMedicalImageProperties, [1314](#)
  - vtkGetMacro, [1307](#), [1308](#)
  - vtkGetObjectMacro, [1308](#), [1309](#)
  - vtkGetStringMacro, [1309](#)
  - vtkSetMacro, [1309](#), [1310](#)
  - vtkSetStringMacro, [1310](#)
  - vtkTypeMacro, [1311](#)
  - Write, [1311](#)
  - WriteGDCMData, [1311](#)
  - WriteSlice, [1311](#)
- vtkGDCMImageWriter.h, [1687](#)
- vtkGDCMMedicalImageProperties, [1312](#)
  - ~vtkGDCMMedicalImageProperties, [1313](#)
  - Clear, [1313](#)
  - GetFile, [1313](#)
  - New, [1313](#)
  - PrintSelf, [1314](#)
  - PushBackFile, [1314](#)
  - vtkGDCMImageReader, [1314](#)
  - vtkGDCMImageReader2, [1314](#)
  - vtkGDCMImageWriter, [1314](#)
  - vtkGDCMMedicalImageProperties, [1313](#)
  - vtkTypeMacro, [1314](#)
- vtkGDCMMedicalImageProperties.h, [1688](#)
- vtkGDCMPolyDataReader, [1315](#)
  - ~vtkGDCMPolyDataReader, [1317](#)
  - FileName, [1319](#)
  - FillMedicalImageInformation, [1317](#)
  - MedicalImageProperties, [1319](#)
  - New, [1317](#)
  - PrintSelf, [1317](#)
  - RTStructSetProperties, [1320](#)
  - RequestData, [1317](#)
  - RequestData\_HemodynamicWaveformStorage, [1318](#)
  - RequestData\_RTStructureSetStorage, [1318](#)
  - RequestInformation, [1318](#)
  - RequestInformation\_HemodynamicWaveformStorage, [1318](#)
  - RequestInformation\_RTStructureSetStorage, [1318](#)
- vtkGDCMPolyDataReader, [1316](#)
- vtkGetObjectMacro, [1318](#), [1319](#)
- vtkGetStringMacro, [1319](#)
- vtkSetStringMacro, [1319](#)
- vtkTypeMacro, [1319](#)
- vtkGDCMPolyDataReader.h, [1688](#)
- vtkGDCMPolyDataWriter, [1320](#)
  - ~vtkGDCMPolyDataWriter, [1322](#)
  - InitializeRTStructSet, [1322](#)
  - MedicalImageProperties, [1324](#)
  - New, [1322](#)
  - PrintSelf, [1323](#)

- RTStructSetProperties, 1325
- SetMedicalImageProperties, 1323
- SetNumberOfInputPorts, 1323
- SetRTStructSetProperties, 1323
- vtkGDCMPolyDataWriter, 1322
- vtkTypeMacro, 1324
- WriteData, 1324
- WriteRTSTRUCTData, 1324
- WriteRTSTRUCTInfo, 1324
- vtkGDCMPolyDataWriter.h, 1689
- vtkGDCMTesting, 1325
  - ~vtkGDCMTesting, 1327
  - GetGDCMDataRoot, 1327
  - GetMD5MetaImage, 1327
  - GetMHDMD5FromFile, 1327
  - GetNumberOfMD5MetaImages, 1327
  - GetRAWMD5FromFile, 1328
  - GetVTKDataRoot, 1328
  - MD5MetaImagesType, 1326
  - New, 1328
  - PrintSelf, 1328
  - vtkGDCMTesting, 1326
  - vtkTypeMacro, 1328
- vtkGDCMTesting.h, 1690
- vtkGDCMThreadedImageReader, 1329
  - ~vtkGDCMThreadedImageReader, 1331
  - ExecuteData, 1331
  - ExecuteInformation, 1331
  - New, 1331
  - PrintSelf, 1331
  - ReadFiles, 1331
  - RequestDataCompat, 1332
  - vtkBooleanMacro, 1332
  - vtkGDCMThreadedImageReader, 1330
  - vtkGetMacro, 1332
  - vtkSetMacro, 1332
  - vtkTypeMacro, 1333
- vtkGDCMThreadedImageReader.h, 1690
- vtkGDCMThreadedImageReader2, 1333
  - ~vtkGDCMThreadedImageReader2, 1335
  - GetFileName, 1335
  - New, 1335
  - PrintSelf, 1335
  - RequestInformation, 1336
  - SetFileName, 1336
  - SetFileNames, 1336
  - SplitExtent, 1336
  - ThreadedRequestData, 1336
  - vtkBooleanMacro, 1337
  - vtkGDCMThreadedImageReader2, 1335
  - vtkGetMacro, 1337, 1338
  - vtkGetObjectMacro, 1339
  - vtkGetVector3Macro, 1339
  - vtkGetVector6Macro, 1339
  - vtkSetMacro, 1339, 1340
  - vtkSetVector3Macro, 1340, 1341
  - vtkSetVector6Macro, 1341
  - vtkTypeMacro, 1341
- vtkGDCMThreadedImageReader2.h, 1691
- vtkGetMacro
  - vtkGDCMImageReader, 1280, 1281
  - vtkGDCMImageReader2, 1295, 1296
  - vtkGDCMImageWriter, 1307, 1308
  - vtkGDCMThreadedImageReader, 1332
  - vtkGDCMThreadedImageReader2, 1337, 1338
  - vtkImageColorViewer, 1353
  - vtkImageMapToColors16, 1361
  - vtkImageMapToWindowLevelColors2, 1365, 1366
- vtkGetObjectMacro
  - vtkGDCMImageReader, 1281, 1282
  - vtkGDCMImageReader2, 1297
  - vtkGDCMImageWriter, 1308, 1309
  - vtkGDCMPolyDataReader, 1318, 1319
  - vtkGDCMThreadedImageReader2, 1339
  - vtkImageColorViewer, 1353, 1354
  - vtkImageMapToColors16, 1361
- vtkGetStringMacro
  - vtkGDCMImageReader, 1282
  - vtkGDCMImageReader2, 1297
  - vtkGDCMImageWriter, 1309
  - vtkGDCMPolyDataReader, 1319
  - vtkRTStructSetProperties, 1384–1386
- vtkGetVector3Macro
  - vtkGDCMImageReader, 1282
  - vtkGDCMImageReader2, 1297
  - vtkGDCMThreadedImageReader2, 1339
- vtkGetVector6Macro
  - vtkGDCMImageReader, 1283
  - vtkGDCMImageReader2, 1297
  - vtkGDCMThreadedImageReader2, 1339
- vtkImageColorViewer, 1342
  - ~vtkImageColorViewer, 1345
  - AddInput, 1345
  - AddInputConnection, 1345
  - FirstRender, 1355
  - GetColorLevel, 1345
  - GetColorWindow, 1346
  - GetInput, 1346
  - GetOffScreenRendering, 1346
  - GetOverlayVisibility, 1346
  - GetPosition, 1346
  - GetSize, 1346
  - GetSliceMax, 1346
  - GetSliceMin, 1347
  - GetSliceRange, 1347
  - GetWindowName, 1347
  - ImageActor, 1355
  - InstallPipeline, 1347

- Interactor, [1355](#)
- InteractorStyle, [1355](#)
- New, [1347](#)
- OverlayImageActor, [1355](#)
- PrintSelf, [1348](#)
- Render, [1348](#)
- RenderWindow, [1356](#)
- Renderer, [1355](#)
- SetColorLevel, [1348](#)
- SetColorWindow, [1348](#)
- SetDisplayId, [1348](#)
- SetInput, [1349](#)
- SetInputConnection, [1349](#)
- SetOffScreenRendering, [1349](#)
- SetOverlayVisibility, [1349](#)
- SetParentId, [1349](#)
- SetPosition, [1349](#), [1350](#)
- SetRenderWindow, [1350](#)
- SetRenderer, [1350](#)
- SetSize, [1350](#)
- SetSlice, [1351](#)
- SetSliceOrientation, [1351](#)
- SetSliceOrientationToXY, [1351](#)
- SetSliceOrientationToXZ, [1351](#)
- SetSliceOrientationToYZ, [1351](#)
- SetWindowId, [1352](#)
- SetupInteractor, [1351](#)
- Slice, [1356](#)
- SliceOrientation, [1356](#)
- UnInstallPipeline, [1352](#)
- UpdateDisplayExtent, [1352](#)
- UpdateOrientation, [1352](#)
- VTK\_LEGACY, [1352](#), [1353](#)
- vtkBooleanMacro, [1353](#)
- vtkGetMacro, [1353](#)
- vtkGetObjectMacro, [1353](#), [1354](#)
- vtkImageColorViewer, [1345](#)
- vtkImageColorViewerCallback, [1355](#)
- vtkTypeMacro, [1354](#)
- WindowLevel, [1356](#)
- vtkImageColorViewer.h, [1691](#)
- vtkImageColorViewerCallback
  - vtkImageColorViewer, [1355](#)
- vtkImageMapToColors16, [1357](#)
  - ~vtkImageMapToColors16, [1358](#)
  - ActiveComponent, [1362](#)
  - DataWasPassed, [1362](#)
  - GetMTime, [1359](#)
  - LookupTable, [1362](#)
  - New, [1359](#)
  - OutputFormat, [1362](#)
  - PassAlphaToOutput, [1363](#)
  - PrintSelf, [1359](#)
  - RequestData, [1359](#)
  - RequestInformation, [1359](#)
  - SetLookupTable, [1359](#)
  - SetOutputFormatToLuminance, [1360](#)
  - SetOutputFormatToLuminanceAlpha, [1360](#)
  - SetOutputFormatToRGBA, [1360](#)
  - SetOutputFormatToRGB, [1360](#)
  - ThreadedRequestData, [1360](#)
  - vtkBooleanMacro, [1360](#)
  - vtkGetMacro, [1361](#)
  - vtkGetObjectMacro, [1361](#)
  - vtkImageMapToColors16, [1358](#)
  - vtkSetMacro, [1361](#), [1362](#)
  - vtkTypeMacro, [1362](#)
- vtkImageMapToColors16.h, [1692](#)
- vtkImageMapToWindowLevelColors2, [1363](#)
  - ~vtkImageMapToWindowLevelColors2, [1364](#)
  - Level, [1366](#)
  - New, [1365](#)
  - PrintSelf, [1365](#)
  - RequestData, [1365](#)
  - RequestInformation, [1365](#)
  - ThreadedRequestData, [1365](#)
  - vtkGetMacro, [1365](#), [1366](#)
  - vtkImageMapToWindowLevelColors2, [1364](#)
  - vtkSetMacro, [1366](#)
  - vtkTypeMacro, [1366](#)
  - Window, [1367](#)
- vtkImageMapToWindowLevelColors2.h, [1692](#)
- vtkImagePlanarComponentsToComponents, [1367](#)
  - ~vtkImagePlanarComponentsToComponents, [1368](#)
  - New, [1368](#)
  - PrintSelf, [1368](#)
  - RequestData, [1369](#)
  - vtkImagePlanarComponentsToComponents, [1368](#)
  - vtkTypeMacro, [1369](#)
- vtkImagePlanarComponentsToComponents.h, [1693](#)
- vtkImageRGBToYBR.h, [1693](#)
- vtkImageRGBToYBR, [1369](#)
  - ~vtkImageRGBToYBR, [1370](#)
  - New, [1371](#)
  - PrintSelf, [1371](#)
  - ThreadedExecute, [1371](#)
  - vtkImageRGBToYBR, [1370](#)
  - vtkTypeMacro, [1371](#)
- vtkImageYBRToRGB.h, [1694](#)
- vtkImageYBRToRGB, [1372](#)
  - ~vtkImageYBRToRGB, [1373](#)
  - New, [1373](#)
  - PrintSelf, [1373](#)
  - ThreadedExecute, [1373](#)
  - vtkImageYBRToRGB, [1373](#)
  - vtkTypeMacro, [1374](#)
- vtkLookupTable16, [1374](#)
  - ~vtkLookupTable16, [1376](#)



- Build, [1376](#)
- GetPointer, [1376](#)
- MapScalarsThroughTable2, [1376](#)
- New, [1376](#)
- PrintSelf, [1377](#)
- SetNumberOfTableValues, [1377](#)
- Table16, [1377](#)
- vtkLookupTable16, [1375](#)
- vtkTypeMacro, [1377](#)
- WritePointer, [1377](#)
- vtkLookupTable16.h, [1694](#)
- vtkRTStructSetProperties, [1378](#)
  - ~vtkRTStructSetProperties, [1380](#)
  - AddContourReferencedFrameOfReference, [1380](#)
  - AddReferencedFrameOfReference, [1380](#)
  - AddStructureSetROIObservation, [1381](#)
  - AddStructureSetROI, [1381](#)
  - Clear, [1381](#)
  - DeepCopy, [1381](#)
  - GetContourReferencedFrameOfReferenceClassUID, [1381](#)
  - GetContourReferencedFrameOfReferenceInstanceClassUID, [1382](#)
  - GetNumberOfContourReferencedFrameOfReferences, [1382](#)
  - GetNumberOfReferencedFrameOfReferences, [1382](#)
  - GetNumberOfStructureSetROIs, [1382](#)
  - GetReferencedFrameOfReferenceClassUID, [1382](#)
  - GetReferencedFrameOfReferenceInstanceUID, [1383](#)
  - GetStructureSetObservationNumber, [1383](#)
  - GetStructureSetROIDescription, [1383](#)
  - GetStructureSetROIGenerationAlgorithm, [1383](#)
  - GetStructureSetROIName, [1383](#)
  - GetStructureSetROINumber, [1383](#)
  - GetStructureSetROIObservationLabel, [1384](#)
  - GetStructureSetROIRefFrameRefUID, [1384](#)
  - GetStructureSetRTROIInterpretedType, [1384](#)
  - Internals, [1388](#)
  - New, [1384](#)
  - PrintSelf, [1384](#)
  - ReferenceFrameOfReferenceUID, [1388](#)
  - ReferenceSeriesInstanceUID, [1388](#)
  - SOPInstanceUID, [1388](#)
  - SeriesInstanceUID, [1388](#)
  - StructureSetDate, [1388](#)
  - StructureSetLabel, [1389](#)
  - StructureSetName, [1389](#)
  - StructureSetTime, [1389](#)
  - StudyInstanceUID, [1389](#)
  - vtkGetStringMacro, [1384–1386](#)
  - vtkRTStructSetProperties, [1380](#)
  - vtkSetStringMacro, [1386, 1387](#)
  - vtkTypeMacro, [1387](#)
- vtkRTStructSetProperties.h, [1695](#)
- vtkSetMacro
  - vtkGDCMImageReader, [1283](#)
  - vtkGDCMImageReader2, [1298](#)
  - vtkGDCMImageWriter, [1309, 1310](#)
  - vtkGDCMThreadedImageReader, [1332](#)
  - vtkGDCMThreadedImageReader2, [1339, 1340](#)
  - vtkImageMapToColors16, [1361, 1362](#)
  - vtkImageMapToWindowLevelColors2, [1366](#)
- vtkSetStringMacro
  - vtkGDCMImageWriter, [1310](#)
  - vtkGDCMPolyDataReader, [1319](#)
  - vtkRTStructSetProperties, [1386, 1387](#)
- vtkSetVector3Macro
  - vtkGDCMThreadedImageReader2, [1340, 1341](#)
- vtkSetVector6Macro
  - vtkGDCMImageReader, [1283](#)
  - vtkGDCMImageReader2, [1298](#)
  - vtkGDCMThreadedImageReader2, [1341](#)
- vtkTypeMacro
  - vtkGDCMImageReader, [1284](#)
  - vtkGDCMImageReader2, [1298](#)
  - vtkGDCMImageWriter, [1311](#)
  - vtkGDCMMedicalImageProperties, [1314](#)
  - vtkGDCMPolyDataReader, [1319](#)
  - vtkGDCMPolyDataWriter, [1324](#)
  - vtkGDCMTesting, [1328](#)
  - vtkGDCMThreadedImageReader, [1333](#)
  - vtkGDCMThreadedImageReader2, [1341](#)
  - vtkImageColorViewer, [1354](#)
  - vtkImageMapToColors16, [1362](#)
  - vtkImageMapToWindowLevelColors2, [1366](#)
  - vtkImagePlanarComponentsToComponents, [1369](#)
  - vtkImageRGBToYBR, [1371](#)
  - vtkImageYBRToRGB, [1374](#)
  - vtkLookupTable16, [1377](#)
  - vtkRTStructSetProperties, [1387](#)
- WLMFindQuery
  - gdcm::WLMFindQuery, [1391](#)
- WarningOff
  - gdcm::Trace, [1117](#)
- WarningOn
  - gdcm::Trace, [1117](#)
- Waveform
  - gdcm::Waveform, [1390](#)
- what
  - gdcm::Exception, [431](#)
- Window
  - vtkImageMapToWindowLevelColors2, [1367](#)
- WindowLevel
  - vtkImageColorViewer, [1356](#)
- Write
  - gdcm::ByteValue, [228](#)
  - gdcm::CSAHeader, [289](#)



gdcm::CommandDataSet, [259](#)  
 gdcm::DataElement, [319](#)  
 gdcm::DataSet, [337](#)  
 gdcm::Element, [396](#)  
 gdcm::Element< TVR, VM::VM1\_n >, [403](#)  
 gdcm::EncodingImplementation< VR::VRASCII >, [422](#)  
 gdcm::EncodingImplementation< VR::VRBINARY >, [423](#)  
 gdcm::ExplicitDataElement, [435](#)  
 gdcm::File, [443](#)  
 gdcm::FileAnonymizer, [448](#)  
 gdcm::FileMetaInformation, [470](#)  
 gdcm::Fragment, [501](#)  
 gdcm::ImageWriter, [578](#)  
 gdcm::ImplicitDataElement, [585](#)  
 gdcm::Item, [605](#)  
 gdcm::PGXCodec, [794](#)  
 gdcm::PNMCodec, [828](#)  
 gdcm::PixmapWriter, [824](#)  
 gdcm::Preamble, [832](#)  
 gdcm::SegmentWriter, [945](#)  
 gdcm::SequenceOfFragments, [952](#)  
 gdcm::SequenceOfItems, [962](#)  
 gdcm::StreamImageWriter, [1019](#)  
 gdcm::SurfaceWriter, [1069](#)  
 gdcm::Tag, [1100](#)  
 gdcm::VRVLSIZE< 0 >, [1271](#)  
 gdcm::VRVLSIZE< 1 >, [1272](#)  
 gdcm::ValueIO, [1247](#)  
 gdcm::VL, [1254](#)  
 gdcm::VR, [1267](#)  
 gdcm::Writer, [1398](#)  
 gdcm::network::AAAbortPDU, [89](#)  
 gdcm::network::AAAssociateACPDU, [94](#)  
 gdcm::network::AAAssociateRJPDU, [96](#)  
 gdcm::network::AAAssociateRQPDU, [102](#)  
 gdcm::network::AResetRPPDU, [126](#)  
 gdcm::network::AResetRQPDU, [128](#)  
 gdcm::network::AbstractSyntax, [106](#)  
 gdcm::network::ApplicationContext, [121](#)  
 gdcm::network::AsynchronousOperationsWindowSub, [133](#)  
 gdcm::network::BasePDU, [175](#)  
 gdcm::network::ImplementationClassUIDSub, [580](#)  
 gdcm::network::ImplementationUIDSub, [581](#)  
 gdcm::network::ImplementationVersionNameSub, [582](#)  
 gdcm::network::MaximumLengthSub, [664](#)  
 gdcm::network::PDataTFPDU, [776](#)  
 gdcm::network::PresentationContextAC, [839](#)  
 gdcm::network::PresentationContextRQ, [847](#)  
 gdcm::network::PresentationDataValue, [851](#)  
 gdcm::network::RoleSelectionSub, [913](#)  
 gdcm::network::SOPClassExtendedNegotiationSub, [995](#)  
 gdcm::network::ServiceClassApplicationInformation, [970](#)  
 gdcm::network::TransferSyntaxSub, [1126](#)  
 gdcm::network::UserInformation, [1240](#)  
 vtkGDCMImageWriter, [1311](#)  
 Write16  
   gdcm::VL, [1254](#)  
 WriteASCII  
   gdcm::Element< TVR, VM::VM1\_n >, [403](#)  
 WriteBuffer  
   gdcm::ByteValue, [229](#)  
   gdcm::SequenceOfFragments, [953](#)  
 WriteBufferAsRGBA  
   gdcm::LookupTable, [654](#)  
 WriteData  
   vtkGDCMPolyDataWriter, [1324](#)  
 WriteFooter  
   gdcm::DictConverter, [364](#)  
 WriteGDCMData  
   vtkGDCMImageWriter, [1311](#)  
 WriteHeader  
   gdcm::DictConverter, [365](#)  
 WriteHelpFile  
   gdcm::BaseQuery, [180](#)  
 WriteImageInformation  
   gdcm::StreamImageWriter, [1020](#)  
 WriteImageSubregionRAW  
   gdcm::StreamImageWriter, [1020](#)  
 WritePointer  
   vtkLookupTable16, [1377](#)  
 WriteQuery  
   gdcm::BaseQuery, [180](#)  
 WriteRTSTRUCTData  
   vtkGDCMPolyDataWriter, [1324](#)  
 WriteRTSTRUCTInfo  
   vtkGDCMPolyDataWriter, [1324](#)  
 WriteRawHeader  
   gdcm::StreamImageWriter, [1020](#)  
 WriteSlice  
   vtkGDCMImageWriter, [1311](#)  
 Writer  
   gdcm::Writer, [1396](#)  
 XMLDictReader  
   gdcm::XMLDictReader, [1401](#)  
 XMLPrinter  
   gdcm::XMLPrinter, [1404](#)  
 XMLPrivateDictReader  
   gdcm::XMLPrivateDictReader, [1408](#)  
 YBR2RGB  
   gdcm::ImageChangePhotometricInterpretation, [530](#)

ZSpacing

gdcm::IPPSorter, [600](#)

ZTolerance

gdcm::IPPSorter, [600](#)